Square-Cut Pizza Sharing is PPA-complete

Argyrios Deligkas* John Fearnley† Themistoklis Melissourgos‡

Abstract

We study the computational complexity of computing solutions for the square-cut pizza sharing problem. In this problem, we have n mass distributions in the plane, and the task is to find a path that uses horizontal and vertical segments that splits each of the masses in half while making at most $n-1$ turns. We show that finding an approximate solution to this problem is PPA-complete, while finding an exact solution is FIXP-hard and in BU. Our PPA-hardness result applies even when all mass distributions are unions of non-overlapping squares, and our FIXP-hardness result applies even when all mass distributions are unions of weighted squares and right-angled triangles. When the path is restricted to have at most $n-2$ turns, we show that the approximate problem becomes NP-complete, and the exact problem becomes ETR-complete.

*Royal Holloway University of London, UK. Email: argyrios.deligkas@rhul.ac.uk
†University of Liverpool, UK. Email: john.fearnley@liverpool.ac.uk
‡Technical University of Munich, Germany. Email: themistoklis.melissourgos@tum.de

Contents

1 Introduction

Mass partition problems ask us to fairly divide measurable objects that are embedded into Euclidean space [28]. Perhaps the most well-known mass partition problem is the *ham sandwich* problem, in which three masses are given in three-dimensional Euclidean space, and the goal is to find a single plane that cuts all three masses in half. Recently, there has been interest in *pizza sharing* problems, which are mass partition problems in the two-dimensional plane. For example, it was recently shown that it is possible to simultaneously bisect four two-dimensional masses using two straight-line cuts [3].

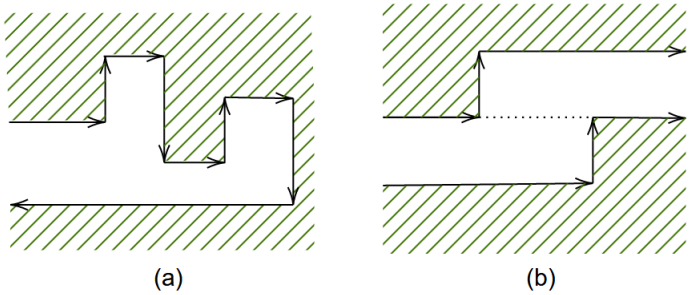


Figure 1: Partitions of the plane to R^+ and R^-. Subfigure (a): a square-cut-path with eight turns. Observe that the path is not y-monotone. Subfigure (b): a y-monotone square-cut-path with four turns.

In this paper, we focus on a different but related problem that we call *square-cut pizza sharing*. In this problem, there are n masses in the $[0,1]^2$ plain. The task is to simultaneously bisect all masses using cuts, but the method of generating the cuts is different. Specifically, we seek a *square-cut*, which consists of a single path that is the union of horizontal and vertical line segments. See Figure 1 for two examples of square-cuts. Intuitively, we can imagine that a pizza cutter is placed on the plane, and is then moved horizontally and vertically without being lifted in order to produce the cut. Note that the path is allowed to wrap around on the horizontal axis: if it exits the left or right boundary, then it re-appears on the opposite boundary. So the cut on the right in Figure 1 is still considered to be a single square-cut.

It has been shown that, given n masses, there always exists a *square-cut-path* (termed SC-*path*) that makes at most $n-1$ turns and simultaneously bisects all of the masses [21]. This holds even if the SC-path is required to be *y-monotone*, meaning that the path never moves downwards [21]. In this paper, we will study the computational complexity of finding such an SC-path.

Computational complexity of fair division problems. There has been much interest recently in the computational complexity of fair division problems. In particular, the complexity class PPA has risen to prominence, because it appears to naturally capture the complexity of solving these problems. For example, it has recently been shown that the ham sandwich problem and the well-known necklace splitting problem are both PPA-complete [14].

More generally, PPA captures the problem of finding a solution to a problem whose solution is guaranteed by the Borsuk-Ulam theorem, since finding an approximate solution to a Borsuk-Ulam function, or finding an exact solution to a linear Borsuk-Ulam function are both known to be PPA-complete problems [27, 7]. The existence of solutions to the ham sandwich problem, the necklace splitting problem, and indeed the square-cut pizza sharing problem can all be proved via the Borsuk-Ulam theorem[1].

Theorem 1 (Borsuk-Ulam)**.** *Let $f : S^d \mapsto \mathbb{R}^d$ be a continuous function, where S^d is a $(d+1)$-dimensional sphere. Then, there exists an $x \in S^d$ such that $f(x) = f(-x)$.*

The other class of relevance here is the class BU, which consists of all problems that can be reduced to finding an *exact* solution to a Borsuk-Ulam function [7]. This class is believed to be substantially

[1]It has also been shown that the Borsuk-Ulam theorem is equivalent to the *ham sandwich theorem* which states that the volumes of any n compact sets in $\mathbb{R}^n$ can always be simultaneously bisected by an $(n-1)$-dimensional hyperplane [29].

harder than the class PPA, because it is possible to construct a Borsuk-Ulam function that only has irrational solutions. Due to this, it is not currently expected that BU will be contained in FNP (which is the class of function problems whose solutions can be verified in polynomial time), whereas the containment of PPA in FNP is immediate.

Unfortunately, it is not currently known whether BU has complete problems. There are, however, problems in BU that are known to be FIXP-hard [7]. FIXP is the class of problems that can be reduced to finding an exact fixed point of a Brouwer function [11]. It is known that FIXP $\subseteq$ BU, and since there exist Brouwer functions that only have irrational fixed points, it is likewise not expected that FIXP will be contained in FNP.

Our contribution. We study the computational complexity of the square-cut pizza sharing problem, and we specifically study the case where all masses are unions of weighted polygons. We study both the exact and the approximate versions of the problem. All of our results are summarized in Table 1.

For the approximate problem, our main result is to show that finding a square cut with $n-1$ turns that ε-approximately bisects n mass distributions is a PPA-complete problem. In fact, we show three different hardness results for the approximate problem.

- Our main hardness result show hardness for $\varepsilon = 1/\text{poly}(n)$, and it is able to show that it is PPA-hard even to find an SC-path with $n + n^{1-\delta}$ turns, where $\delta < 1$ is a constant.
- By making adjustments to the result above, we are able to extend the hardness to the case where all of the polygons are axis-aligned unweighted non-overlapping squares.
- Finally, we are able to show a hardness result for constant ε, but in this case the hardness is PPAD-hardness, which is a slightly weaker hardness result than PPA-hardness.

We then turn our attention to the computational complexity of finding an *exact* solution to the problem. Here we show that the problem of finding an SC-path with at most $n-1$ turns that exactly bisects n masses lies in BU, and is FIXP-hard. This hardness result applies even if all mass distributions are unions of weighted axis-aligned squares and right-angled triangles. In order to prove this result, we provide a simpler existence proof for a solution to the square-cut pizza sharing problem that follows the lines of the original proof from [21].

Finally, we study the decision version of the problem. While a solution to the problem is guaranteed to exist for cuts that make $n-1$ turns, this is not the case if only $n-2$ turns are allowed. We show that deciding whether there exists an approximate solution that makes at most $n-2$ turns is NP-complete, and deciding whether there is an exact solution that makes at most $n-2$ turns is ETR-complete, where ETR consists of every decision problem that can be formulated in the existential theory of the reals.

From a technical point of view, our containment results are shown by directly reducing the square-cut pizza sharing problem to the Borsuk-Ulam problem. Our hardness results are obtained by reducing from the *consensus halving* problem, which was one of the first fair-division problems to be shown to be PPA-complete [13]. We provide a single reduction from consensus halving to square-cut pizza sharing, and then the various different hardness results for consensus halving yield the different results that we have stated above. We remark that, if in the future, consensus halving is shown to be BU-complete under block and triangle valuations for the agents, then our work will also imply that exact square-cut pizza sharing will also be BU-complete.

The technical parts of the proofs are deferred to the Appendix.

Further related work. Since mass partitions lie in the intersection of topology, discrete geometry, and computer science there are several surveys on the topic; [4, 6, 24, 32] focus on the topological point of view, while [1, 9, 19, 20, 22] focus on computational aspects. Consensus halving [31] is the mass partition problem that received the majority of attention in Theoretical Computer Science so far [8, 12, 14, 15, 16].

2 Preliminaries

Mass distributions. A *mass distribution* μ on $[0,1]^2$ is a measure on the plane such that all open subsets of $[0,1]^2$ are measurable, $0 < \mu(\mathbb{R}^2) < \infty$, and $\mu(S) = 0$ for every subset of $[0,1]^2$ with

Reduction	Hardness	ε	Turns	Pieces	Overlap	Theorem
Overlapping	PPA	$\frac{1}{\text{poly}(n)}$	$n + n^{1-\delta}$	$\text{poly}(n)$	$O(n)$	11
Overlapping	PPAD	c	$n + k$	6	3	12
Overlapping	NP	$\frac{1}{\text{poly}(n)}$	$n - 2$	6	3	13
Checkerboard	PPA	$\frac{1}{\text{poly}(n)}$	$n + n^{1-\delta}$	$\text{poly}(n)$	0	14
Exact	FIXP	0	$n - 1$	6	3	16
Exact	ETR	0	$n - 2$	6	3	17

Table 1: A summary of our hardness results. Here, c, k, and δ are absolute constants. Turns denotes the maximum number of turns the path can have. Pieces refers to the maximum number of distinct polygons that define every mass distribution. Overlap denotes the number of different mass distributions that can contain any point of $[0,1]^2$.

dimension lower than 2. A mass distribution μ is *finite-separable*, or simply separable, if it can be decomposed into a finite set of non-overlapping areas $a_1, a_2, \ldots, a_d$ such that $\mu([0,1]^2) = \sum_{i=1}^{d} \mu(a_i)$. In addition, a separable mass distribution μ is *piece-wise uniform*, if for every i and every $S \subseteq a_i$ it holds that $\mu(a_i \cap S) = c_i \cdot \text{area}(a_i \cap S)$ for some $c_i > 0$ independent of S. A separable mass distribution μ is *uniform* if for every $S \subseteq a_i$ it holds that $\mu(a_i \cap S) = \text{area}(a_i \cap S)$. Finally, a mass distribution is *normalised* if $\mu([0,1]^2) = 1$. A set of mass distributions has *overlap* k if any point of $[0,1]^2$ belongs to at most k distributions.

Mass distributions can be categorized according to their shape as well. So, a separable mass distribution is a:

- d**-polygon**, if it can decomposed into d non-overlapping polygons $p_1, p_2, \ldots, p_d$;
- d-ℓ**-square**, if it can decomposed into d squares $s_1, s_2, \ldots, s_d$ each of edge-size ℓ, when $\ell = 1$ we say that it is a unit-square.

We will denote by $\widehat{ABC}$ a triangle with vertices A, B, C and, when clear from context, we will also use the same notation to indicate the *area* of the triangle. Two intersecting line segments AB, BC define two *angles*, denoted $\widehat{ABC}$ and $\widehat{CBA}$. The order of the vertices implies a direction of the segments, i.e. in the former angle we have AB, BC and in the latter we have CB, CA. We consider the direction of the segments and define the angle to be the intersection of the *left* halfspaces of the segments. Therefore $\widehat{ABC} = 360^\circ - \widehat{CBA}$. This order will not matter if clear from context (e.g. in triangles).

Square-cut-paths. A *square-cut-path*, denoted for brevity SC-path, is a non-crossing directed path that is formed only by horizontal and vertical line segments and in addition it is allowed to "wrap around" in the horizontal dimension. Figure 1 shows two examples of SC-paths. A *turn* of the path is where a horizontal segment meets with a vertical segment. An SC-path is *y-monotone* if all of its horizontal segments are monotone with respect to y axis. Any SC-path naturally partitions the plane into two regions, that we call R^+ and R^-.

Pizza sharing. An SC-path ε-*bisects* a mass distribution μ, if $|\mu(R^+) - \mu(R^-)| \leq \varepsilon$ and it simultaneously ε-bisects a set of mass distributions M if $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$ for every $\mu_i \in M$. The ε-SC-Pizza-Sharing problem is to find a SC-path with $n - 1$ turns that simultaneously ε-bisects a set of n mass distributions.

Definition 2. *For any $n \geq 1$, the problem ε-SC-Pizza-Sharing is defined as follows:*

- ***Input:*** *$\varepsilon > 0$ and mass distributions $\mu_1, \mu_2, \ldots, \mu_n$ on $[0,1]^2$.*
- ***Output:*** *A partition of $[0,1]^2$ to R^+ and R^- using a y-monotone SC-path with at most $n - 1$ turns such that for each mass distribution i it holds that $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$.*

In [21] it was proven that ε-SC-Pizza-Sharing always admits a solution for every $\varepsilon \geq 0$. Although it is not hard to construct instances of ε-SC-Pizza-Sharing where $n - 1$ turns are necessary for any

SC-path in order to define a solution, there might be cases where a solution can be achieved with an SC-path with fewer turns. Hence, study the decision version of the problem, denoted ε-SC-Pizza-Sharing(D), where we ask whether we can find a solution with k turns, where $k < n - 1$.

Complexity classes. ε-SC-Pizza-Sharing is an example of a *total* problem, which is a problem that always has a solution. The complexity class TFNP (Total Function NP) [25] contains all total problems whose solutions can be verified in polynomial time.

There are several well-known subclasses of TFNP that we will use in this paper. The class PPA, defined in [27], captures problems whose totality is guaranteed by the *parity argument* on undirected graphs. The complexity class PPAD $\subseteq$ PPA is the subclass of PPA containing all problems whose totality is guaranteed by the parity argument on *directed* graphs. We will show hardness and completeness results for these classes by reducing from the consensus halving problem, which is discussed below.

The complexity class ETR consists of all decision problems that can be formulated in the *existential theory of the reals* [23, 30]. It is known that NP $\subseteq$ ETR $\subseteq$ PSPACE [5], and it is generally believed that ETR is distinct from the other two classes. The class FETR (Function ETR) consists of all search problems whose decision version is in ETR.

We also use two complexity classes that can be the thought of as capturing total search problems within ETR, or to be more specific, TFETR (Total Function ETR).[2] The class BU $\subseteq$ TFETR was introduced in [7] and captures problems whose totality is guaranteed by the Borsuk-Ulam theorem. The class FIXP $\subseteq$ BU was defined in [10] and captures problems whose totality is guaranteed by Brouwer's fixed point theorem.

Consensus Halving. The hardness results that we will show in this paper will be shown by a reduction from the consensus halving problem.

In the ε-Consensus-Halving problem, there is a set of n agents with *valuation functions* v_i over the interval $[0, 1]$, and the goal is to find a partition of the interval into subintervals labelled either "$+$" or "$-$", using at most n cuts. This partition should satisfy that for every agent i, the total value for the union of subintervals $\mathcal{I}^+$ labelled "$+$" and the total value for the union of subintervals $\mathcal{I}^-$ labelled "$-$" is the same up to ε, i.e., $|v_i(\mathcal{I}^+) - v_i(\mathcal{I}^-)| \leq \varepsilon$. We will consider the following types for a valuation function v_i; see Figure 2 for a visualization.

- *k-block.* v_i can be decomposed into at most k non-overlapping (but possibly adjacent) intervals $[a_{i1}^\ell, a_{i1}^r], \ldots, [a_{ik}^\ell, a_{ik}^r]$ where interval $[a_{ij}^\ell, a_{ij}^r]$ has density c_{ij} and 0 otherwise. So, $v_i([a_{ij}^\ell, x]) = (x - a_{ij}^\ell) \cdot c_{ij}$ for every $x \in [a_{ij}^\ell, a_{ij}^r]$ and $v_i([0,1]) = \sum_j v_i([a_{ij}^\ell, a_{ij}^r]) = 1$.
- *2-block uniform.* v_i is two-block and the density of every interval is c_i.
- *k-block-triangle.* v_i is the union of a k-block valuation function and an extra interval $[a_{i1}^\ell, a_{i1}^r]$, where $v_i([a_{i1}^\ell, x]) = (a_{i1}^\ell - x)^2$ for every $x \in [a_{i1}^\ell, a_{i1}^r]$ and $(a_{i1}^\ell, a_{i1}^r) \cap [a_{ij}^\ell, a_{ij}^r] = 0$ for every $j \in [k]$.

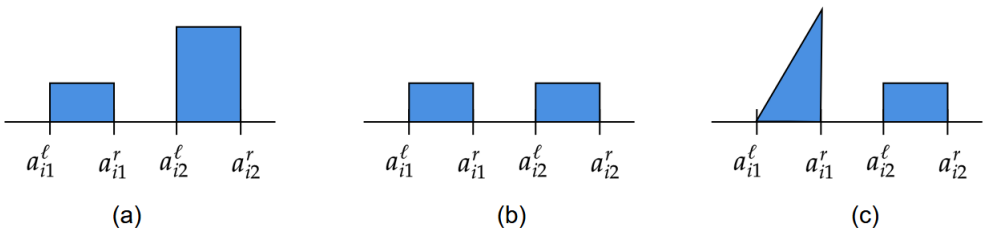


Figure 2: (a) 3-block valuation; (b) 2-block uniform valuation; (c) 1-block-triangle valuation.

3 Containment results for SC-Pizza-Sharing

In this section we present containment results for the exact and approximate versions of SC-Pizza-Sharing that we study in this paper. All of our containment results revolve around a proof that

[2]Both FETR and TFETR were introduced in [7] as the natural analogues of FNP and TFNP.

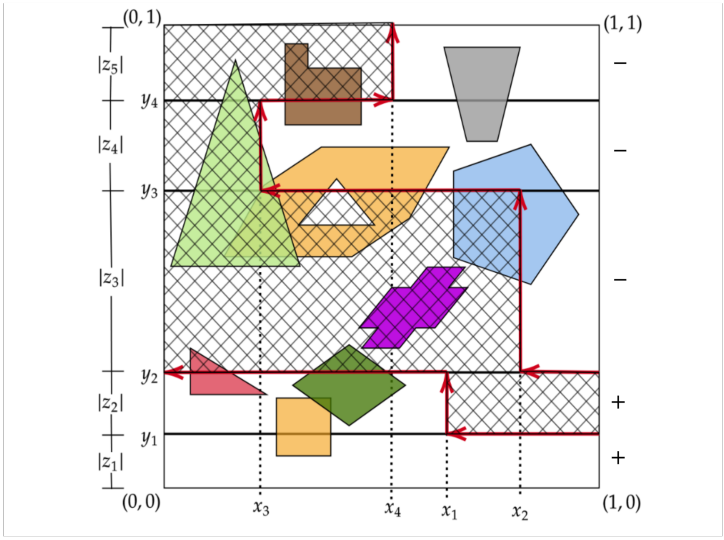


Figure 3: An instance with 8 mass distributions. A vector on S^8 corresponds to slices and vertical cuts, which define a y-monotone SC-path.

solutions exist for SC-Pizza-Sharing that utilizes the Borsuk-Ulam theorem. A proof of this kind was already presented in [21], but we will present our own proof which is conceptually simpler, as it does not use any involved topological techniques. An advantage of our proof is that it can be made algorithmic, and so it can be used to show that SC-Pizza-Sharing is contained in BU. Then, by making simple modifications to the BU containment proof, we show the other containment results hold.

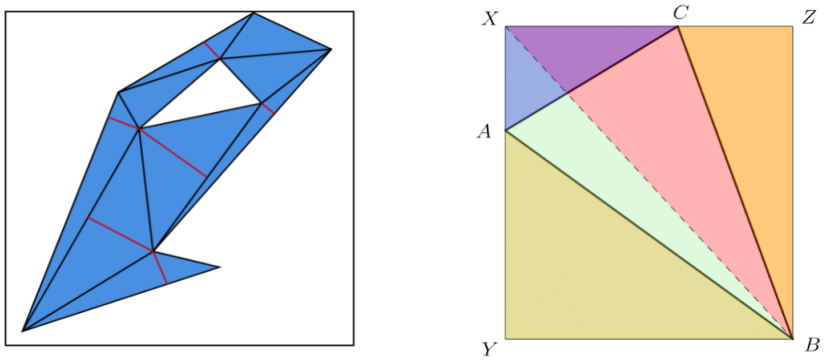
3.1 Containment results for exact SC-Pizza-Sharing

Existence of a SC-Pizza-Sharing solution. We begin by proving that a solution to exact SC-Pizza-Sharing always exists. This proof holds for arbitrary mass distributions, but for our algorithmic results we will only consider the case where the mass distributions are unions of polygons with holes. Our proof is based on the proof of Karasev, Roldán-Pensado and Soberón given in [21], but they use more involved techniques from topology, which we would like to avoid, since our goal is to implement the result algorithmically.

Let S^n denote the L_1 sphere in $(n+1)$-dimensions. The Borsuk-Ulam theorem states that if $f : S^n \mapsto \mathbb{R}^n$ is a continuous function, then there exists a point $\vec{P} \in S^n$ such that $f(\vec{P}) = f(-\vec{P})$. We will show how to encode SC-paths as a points in S^n, and then we will build a function f that ensures that $f(\vec{P}) = f(-\vec{P})$ only when the SC-path corresponding to $\vec{P}$ is a solution to the SC-Pizza-Sharing problem.

Figure 3 gives an overview for our SC-path embedding. The embedding considers only y-monotone SC-paths. The path itself is determined by the points $x_1, x_2, \ldots$, and $y_1, y_2, \ldots$, which define the points at which the path turns. The path begins on the boundary on the line y_1, and then moves to x_1, at which points it turns and moves upwards to y_2, and it then turns to move to x_2, and so on.

But these points alone do not fully specify the path, since the decision of whether to move to the right or to the left when moving from x_i to x_{i+1} has not been specified. To do this, we also include signs that are affixed to each horizontal strip. We will call the two sides of the cut side A (non-shaded) and side B (shaded). The $(i+1)$-st strip $[y_i, y_{i+1}]$ is split into two by the vertical line segment on $x = x_i$ that starts from point (x_i, y_i) and ends at (x_i, y_{i+1}), with one part being on side A, and the other being on side B. If the strip is assigned the sign $+$ then the area on the left of the strip is on side A of the cut, while if the strip is assigned the sign $-$ then the areas on the right of the strip is on side A of the cut. Once the sides of the cut have been decided, there is then a unique way to move through the points in order to separate sides A and B.



(a) (b)

Figure 4: (a) The triangulation with only non-obtuse triangles. After the standard triangulation, extra line segments (in red colour) are added to ensure non-obtuseness. (b) A non-obtuse triangle $\widehat{ABC}$ and its decomposition into axis-aligned right-angled triangles: $\widehat{XYB} + \widehat{XBZ} - \widehat{AYB} - \widehat{XAC} - \widehat{CBZ}$.

So, an SC-path with $n-1$ turns, can be represented by n variables and $\lceil n/2 \rceil$ signs. We then embed these into the sphere S^n in the following way. We give an informal description here, and the full definition will be given in the proof of Theorem 3. We encode the y values as variables $z_i \in [-1,1]$ where $y_{i+1} = y_i + |z_i|$, while the x values are encoded as values in $[-1,1]$, where $|x_i|$ defines the ith value on the x axis. We use the signs of the z_i variables to define the signs for the strips. Finally, we then shrink all variables so that their sum lies in $[-1,1]$, and we embed them as a point in S^n using one extra dimension as a slack variable in case the sum of the absolute values of the variables does not equal 1.

The key property is that, if a point $\vec{P} \in S^n$ represents an SC-path, then the point $-\vec{P}$ represents exactly the same SC-path but with all signs flipped, and thus sides A and B will be exchanged. Therefore, we can write down a function f, where $f(\vec{P})_i$ outputs the amount of mass of the i-th mass distribution that lies on the A side of the cut, and therefore any point $\vec{P}$ satisfying $f(\vec{P}) = f(-\vec{P})$ must cut all mass distributions exactly in half.

So, we get the following theorem that is proved formally in Appendix B.

Theorem 3 (originally by [21])**.** *Let n be a positive integer. For any n mass distributions in $\mathbb{R}^2$, there is a path formed by only horizontal and vertical segments with at most $n-1$ turns that splits $\mathbb{R}^2$ into two sets of equal size in each measure. Moreover, the path is y-monotone.*

BU-containment. The next step is to turn this existence result into a proof that SC-Pizza-Sharing is contained in BU. We begin by recapping the definition of BU given in [7]. An arithmetic circuit is a circuit that operates on real numbers, and uses gates from the set $\{c, +, -, \times c, \times, \max, \min\}$, where a c-gate outputs the constant c, a $\times c$ gate multiplies the input by a constant c, and all other gates behave according to their standard definitions. The class BU contains every problem that can be reduced to Borsuk-Ulam.

Definition 4 (Borsuk-Ulam)**.**

- **Input:** *A continuous function $f : \mathbb{R}^{d+1} \mapsto \mathbb{R}^d$ presented as an arithmetic circuit*
- **Task:** *Find an $x \in S^d$ such that $f(x) = f(-x)$.*

While Theorem 3 utilizes the Borsuk-Ulam theorem, it does not directly show containment in BU, because it does not construct an arithmetic circuit. We now show how this can be done if the mass distributions are unions of polygons with holes.

The key issue is how to determine how much of a polygon lies on a particular side of the cut. To do this, we first triangulate all polygons, as shown in Figure 4(a) to obtain mass distributions that

are the unions of triangles. We then break each individual triangle down into a sum of axis-aligned right-angled triangles. This is shown in Figure 4(b), where the triangle $\widehat{ABC}$ is represented as the sum of $\widehat{XYB} + \widehat{XBZ} - \widehat{AYB} - \widehat{XAC} - \widehat{CBZ}$.

We then explicitly build an arithmetic circuit that, given the point $x \in S^{n+1}$ used in Theorem 3, and a specific axis-aligned right-angled triangle, can output the amount of mass of that triangle that lies on side A of the cut. Then the function f used in Theorem 3 can be built simply by summing over the right-angled triangles that arise from decomposing each of the polygons. So we get the following theorem, which is proved formally in Appendix C.

Theorem 5. *Exact* SC-Pizza-Sharing *for weighted polygons with holes is in* BU.

ETR containment. In the following theorem we show that the problem of deciding whether there exists an exact solution of SC-Pizza-Sharing with n mass distributions and $k \in \mathbb{N}$ turns in the SC-path is in ETR. Consequently, this implies that the respective search problem lies in FETR, and as it is apparent from the BU containment result of this paper, when $k \geq n-1$ the problem is in BU ($\subseteq$ TFETR $\subseteq$ FETR). To show this we use the proof of Theorem 5. The detailed proof can be found in Appendix D.

Theorem 6. *For any $k \in \mathbb{N}$, deciding whether there exists an* SC*-path with k turns that is an exact solution for* SC-Pizza-Sharing *with n mass distributions is in* ETR.

3.2 Containment results for approximate SC-Pizza-Sharing

PPA containment. The following theorem shows PPA containment of ε-SC-Pizza-Sharing via a reduction to the ε-Borsuk-Ulam problem which is in PPA [7].

Definition 7 (ε-Borsuk-Ulam)**.**

- **Input:** *A continuous function $f : \mathbb{R}^{d+1} \mapsto \mathbb{R}^d$ presented as an arithmetic circuit, along with two constants $\varepsilon, \lambda > 0$.*
- **Task:** *Find one of the following.*
  1. *A point $x \in S^d$ such that $\|f(x) - f(-x)\|_\infty \leq \varepsilon$.*
  2. *Two points $x, y \in S^d$ such that $\|f(x) - f(y)\|_\infty > \lambda \cdot \|x - y\|_\infty$.*

If the second task is accomplished, then we have found witnesses $x, y \in S^d$ that function f is not λ-Lipschitz continuous in the L_∞-norm as required. But if the first task is accomplished then we have an approximate solution to the Borsuk-Ulam problem. To prove the theorem we utilize Theorem 5 and the Lipschitzness of the function we have constructed in its proof. The detailed proof can be found in Appendix E.

Theorem 8. ε-SC-Pizza-Sharing *for weighted polygons with holes is in* PPA.

NP containment. Finally, we show deciding whether there exists a solution for ε-SC-Pizza-Sharing with k turns is in NP, for any $k \in \mathbb{N}$. We prove this via a reduction to ε-Borsuk-Ulam problem when ε-SC-Pizza-Sharing is parameterized by the number of turns the SC-path must have. Our proof combines ideas and results from the proofs of Theorems 6, 8, and 5. The detailed proof can be found in Appendix F.

Theorem 9. *For any $k \in \mathbb{N}$, deciding whether there exists an* SC*-path with k turns that is a solution for* ε-SC-Pizza-Sharing *with n mass distributions is in* NP.

4 Hardness results for SC-Pizza-Sharing

Here we show all hardness results regarding the exact and approximate versions of SC-Pizza-Sharing. In Section 4.1 we provide hardness reductions for ε-SC-Pizza-Sharing, while in Section 4.2 we give hardness reductions for exact SC-Pizza-Sharing.

4.1 Hardness results for approximate SC-Pizza-Sharing

In this section we prove several hardness results for ε-SC-Pizza-Sharing. We give two different reductions that allow us to prove a variety of results. The first one, which we call the "overlapping" reduction, is conceptually simpler, and it reduces from ε-Consensus-Halving with k-block valuations. It produces SC-Pizza-Sharing instances where every mass distribution is a union of piece-wise uniform unit-squares. We use this to prove PPA-hardness, PPAD-hardness, and NP-hardness for the problem when we have overlapping mass distributions.

The second reduction, which we call the "checkerboard" reduction is more technical and reduces from ε-Consensus-Halving with 2-block uniform valuations. It allows us to prove PPA-hardness even when every mass distribution is d_i-ℓ_i-unit-square uniform. While the overlapping reduction produces mass distributions that can overlap each other, the checkerboard reduction produces an instance in which the mass distributions do not touch each other.

Since both reductions are from ε-Consensus-Halving with block valuations, we will begin by introducing some notation. For both reductions, we will reduce from an instance I_{CH} of ε-Consensus-Halving with k-block valuations for the agents. Thus, the valuation function of agent i is defined by the subintervals $[a_{ij}^{\ell}, a_{ij}^{r}]$ for $j \in [k]$.

The first step of both reductions is to partition $[0,1]$ to subintervals that are defined by *points of interest*. We say that a point $x \in [0,1]$ is a point of interest if it coincides with the beginning or the end of a valuation block of an agent; formally, x is a point of interest if $x \in \{a_{ij}^{\ell}, a_{ij}^{r}\}$ for some $i \in [n]$ and $j \in [k]$. Let $0 \leq x_1 \leq x_2 \leq \ldots \leq x_m \leq 1$ denote the points of interest and $[x_j, x_{j+1}]$ denote the intervals of interest. Observe that $m \leq 2 \cdot n \cdot k$. In addition, observe that for every j the difference $x_{j+1} - x_j$ is polynomially bounded with respect to the input size.

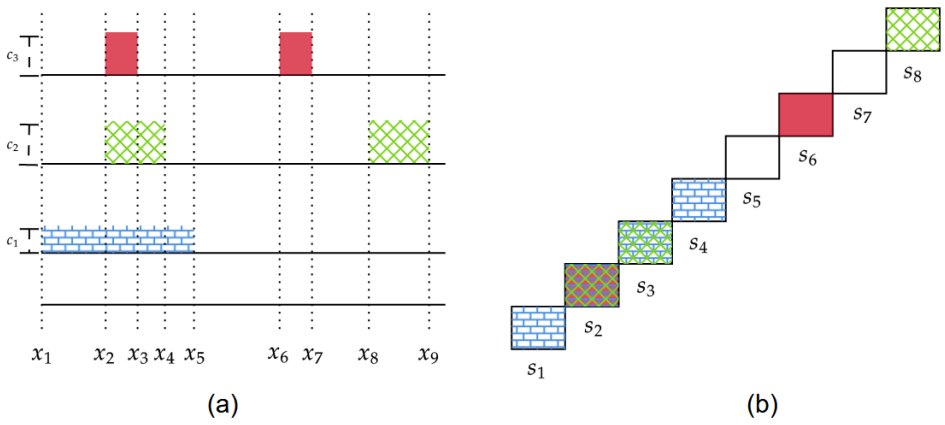


Figure 5: Subfigure (a): An instance of Consensus-Halving with three agents with block uniform valuations and the corresponding points of interest. Subfigure (b): the corresponding instance of SC-Pizza-Sharing; observe that on s_2 all three different mass distributions exist and on s_3 the green and the blue distributions exist.

Overview of the overlapping reduction. The key idea behind the reduction is to encode Consensus-Halving instance as a sequence of diagonal squares in the SC-Pizza-Sharing instance. This can be seen in Figure 5. On the left we show a Consensus-Halving instance with three agents. The points of interest, which consist of any point at which some agent's valuation changes, are shown along the bottom of this instance. These points conceptually splits the Consensus-Halving instance into *blocks*, since in between any pair of consecutive points of interest, all agents have a non-changing valuation.

On the right in Figure 5 we show how this instance is encoded in the SC-Pizza-Sharing instance. There are three mass distributions, coloured blue, green, and red, and each mass distribution corresponds to an agent from the Consensus-Halving instance. The instance consists of a sequence of squares, which are lined up diagonally, and each square corresponds to the corresponding block from the Consensus-Halving instance. Square s_1 corresponds to the region between x_1 and x_2 and so contains a single block from the blue mass distribution, square s_2 corresponds to the region between x_2

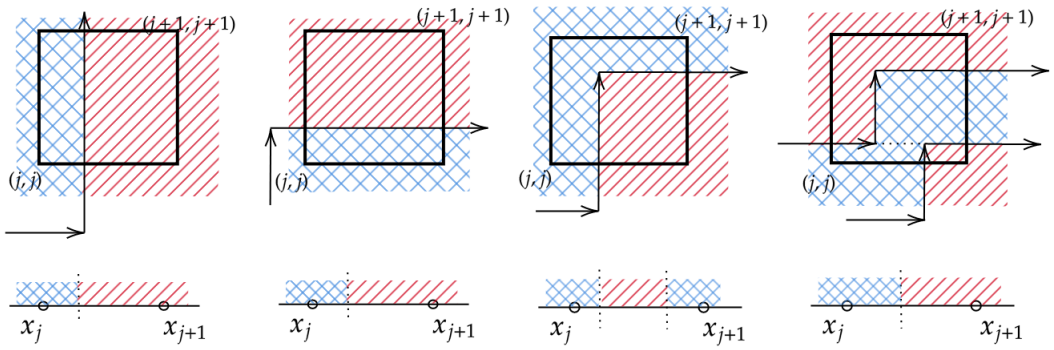


Figure 6: Translation of the turns of the SC-path to cuts. Observe that we need one turn in order the SC-path to enter the square.

and x_3 and so contains three blocks, and so on. The squares are weighted according to the valuations of the agents, so in s_2, the red and green squares get higher weights than the blue square, since the red and green agents value that block higher than the blue agent does.

The key observation is that if an SC-path cuts two distinct squares, then it must make at least one turn in between due to the diagonal arrangement of the squares. This will allow us to transform an SC-path with $n-1$ turns into a sequence of n cuts for the Consensus-Halving instance: every time the path passes through a square, we transform it into a cut in the corresponding block in the Consensus-Halving instance. The weighting of the squares ensures that, if the SC-path cuts each of the mass distributions in half, then the cuts will be a solution to the Consensus-Halving instance.

The construction. We are ready to formally define the reduction. Starting from an instance I_{CH} of Consensus-Halving, we will construct an ε-SC-Pizza-Sharing instance I_{SC}.

For every agent i we will create a mass distribution μ_i. We will create $m-1$ unit-squares $s_1, s_2, \ldots, s_{m-1}$ which will be the locations where the mass distributions are placed. Unit-square s_j is defined by the points (j, j), $(j+1, j)$, $(j, j+1)$, and $(j+1, j+1)$, meaning that squares s_j and s_{j+1} are diagonally adjacent.

If agent i has positive value c_{ij} over the interval of interest $[x_j, x_{j+1}]$, then mass distribution μ_i has density $c_{ij} \cdot (x_{j+1} - x_j)$ over the j-th unit-square. This means that for any mass distribution that appears in square s_j and any area of size $S \in [0,1]$ in square s_j, we have that $\mu_i(S) = S \cdot c_i \cdot (x_{j+1} - x_j)$.

Observe that by the way the instance I_{SC} is constructed, any SC-path with k turns can traverse at most $k+1$ squares. We must now show how the SC-path can be mapped to a solution for I_{CH}.

Figure 6 shows how we translate an SC-path into a sequence of cuts for I_{CH}. The first two diagrams show the simple case where the SC-path passes through a square without turning. This can simply be mapped back to a single cut in I_{CH} that cuts the same proportion of mass from the block.

If the SC-path turns inside a square, then we must be more careful. In the third diagram the path turns inside the square, and the square immediately before and the square immediately after the square are both on the same side of the cut. This means that we cannot use a single cut in I_{CH}, because then one of the two adjacent blocks would be in the red region. So instead, we use two cuts. This is not a problem, because the extra turn inside the square means that we can afford to use an extra cut inside this block in I_{CH}.

Fortunately, even if the path passes through the square multiple times, we never need to use more than two cuts in I_{CH}. In general, if the lower left corner of the square and the upper right corner of the square are on the same side of the SC-path, then we use two cuts in I_{CH} then we can use one cut. The final example in Figure 6 shows that, even though the SC-path passes through the square multiple times, we can still map this back to a single cut in I_{CH} because the lower left and upper right corners of the square are on opposite sides of the path. Formally, the translation between SC-paths and cuts in I_{CH} is given in the following lemma, whose proof appears in Appendix G.

Lemma 10. *Every solution of the* ε-SC-Pizza-Sharing *instance* I_{SC} *with an* SC*-path with* k *turns corresponds*

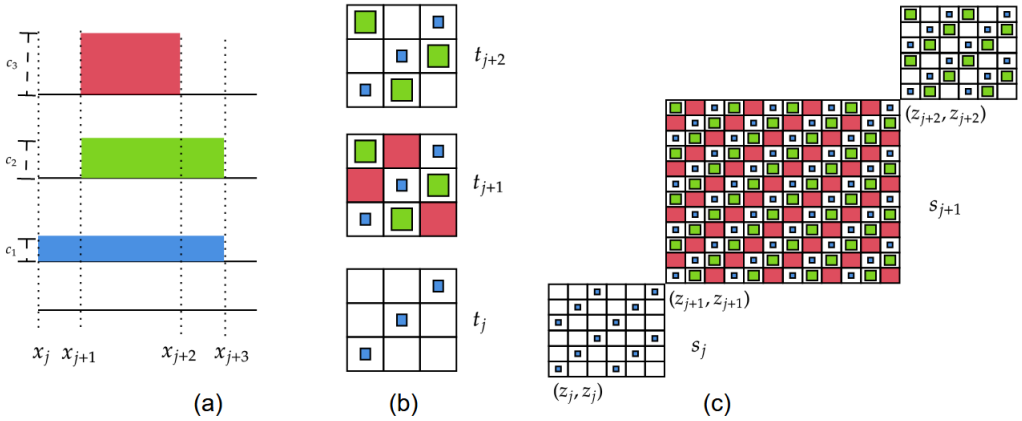


Figure 7: Subfigure (a): a part of the Consensus-Halving instance with three agents and the corresponding regions of interest. Here $x_{j+1} - x_j = x_{j+3} - x_{j+2}$ and $x_{j+2} - x_{j+1} = 4 \cdot (x_{j+3} - x_{j+2})$. Subfigure (b): the corresponding tiles. Subfigure (c): the blocks B_j, B_{j+1} and B_{j+2}. Observe that the size of B_{j+1} is four times larger than the size of B_j.

to a solution of the ε-Consensus-Halving *instance* I_{CH} *with* $k+1$ *cuts.*

Hardness results. So far we have reduced Consensus-Halving to ε-SC-Pizza-Sharing, but the hardness result that we obtain depends on the instance of Consensus-Halving that we start with. We will show that, by varying this instance, we can obtain a variety of hardness results.

We begin by showing PPA-hardness. In [15] it was proven that ε-Consensus-Halving is PPA-hard for $\varepsilon = 1/\text{poly}(n)$ even when we are allowed to use $n + n^{1-\delta}$ cuts, for some constant $\delta > 0$. This result, combined with Lemma 10, gives us the following.

Theorem 11. ε-SC-Pizza-Sharing *is* PPA*-hard for* $\varepsilon = 1/\text{poly}(n)$*, even when an* SC*-path with* $n + n^{1-\delta}$ *turns is allowed for some constant* $\delta > 0$*, and every mass distribution is piece-wise uniform over polynomially many unit-squares.*

Next we can show PPAD-hardness. In [12] it was proven that ε-Consensus-Halving is PPAD-hard for a *constant* ε, 6-block valuation functions, and even when we are allowed to use $n + k$ cuts, for some constant k. In addition, every agent has positive value for at most six intervals of interest and for every interval of interest at most three agents have positive value. So, combining this with Lemma 10 gives us the following.

Theorem 12. ε-SC-Pizza-Sharing *is* PPAD*-hard for some absolute constant* ε*, even when an* SC*-path with* $n + k$ *turns is allowed, for some constant* $k > 0$*, every mass distribution is piece-wise uniform over six unit-squares, and at any point of the plane at most three mass-distributions overlap.*

Finally we can show an NP-hardness result. The instances from [12] were used to prove that it is NP-hard to decide whether ε-Consensus-Halving admits a solution with $n - 1$ cuts. Again, we can use Lemma 10 to get the following.

Theorem 13. *It is* NP*-hard to decide if an* ε-SC-Pizza-Sharing *instance admits a solution with an* SC*-path with* $n - 2$ *turns, even when every mass distribution is piece-wise uniform over six unit-squares, and at any point of the plane at most three mass-distributions overlap.*

The checkerboard reduction. Finally, we show that the problem remains PPA-hard even for unweighted non-overlapping squares. More formally, this means that the i-th mass distribution consists of d_i squares of size $\ell_i \times \ell_i$ each and μ_i is uniformly distributed over the d_i squares. In addition, there is no overlap between the mass distributions.

To do this, we use the same ideas as the overlapping reduction, but rather than producing overlapping squares, we create blocks instead that are filled with a checkerboard pattern to ensure that

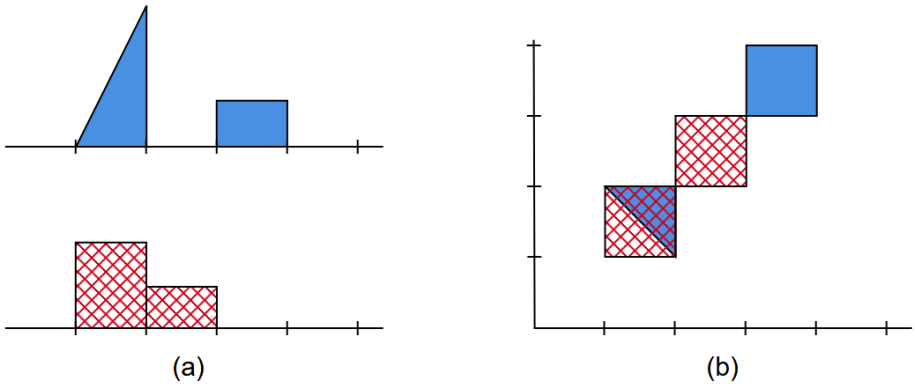


Figure 8: Subfigure (a): a part of the Consensus-Halving instance with two agents and the corresponding regions of interest. Subfigure (b): the corresponding part in the SC-Pizza-Sharing instance.

none of the mass distributions overlap. This is shown in Figure 7. The diagram on the left shows a Consensus-Halving instance, and the diagram on the right shows the corresponding SC-Pizza-Sharing instance. Note that now each block is filled by a tiling that contains squares from the mass distributions that appear in the corresponding interval of interest, where the tiling itself is shown in the middle figure. This ensures that no two mass distributions overlap.

The size of the squares associated with a specific mass distribution that appear in a tile are in proportion to the agent's valuations in the Consensus-Halving instance. Since the squares in the tiles have different sizes, SC-paths in this instance do not perfectly map to cuts in the Consensus-Halving instance, since, for example, a different proportion of the red and blue masses may be cut within a single square. However, we are able to show that with a sufficiently dense tiling, the difference in proportions can be bounded, and so we can still recover an ε-solution for Consensus-Halving from an ε'-solution for SC-Pizza-Sharing. The following theorem is proved formally in Appendix H.

Theorem 14. *ε-SC-Pizza-Sharing is PPA-hard even when ε is inverse-polynomial with respect to n, one is allowed to use $n + n^{1-\delta}$ turns for some constant $\delta > 0$, every mass distribution μ_i is uniform d-ℓ_i-square with $d = O(n)$, and there is no overlap between any two mass distributions.*

4.2 Hardness results for exact SC-Pizza-Sharing

In this section we show hardness results for *exact* SC-Pizza-Sharing. We prove that solving SC-Pizza-Sharing is FIXP-hard and that deciding whether there exists a solution for SC-Pizza-Sharing with strictly less than $n-1$ turns is ETR-hard. We prove these results using a reduction that we call the "exact" reduction, which again starts from Consensus-Halving. This time however we will use the instances of Consensus-Halving produced in [7], which we will denote by I_{CH}^{DFMS}. We note that here the input consists of sets of points, i.e. we describe polygons by their vertices (see Appendix A). In [7] the FIXP-hard family of instances we reduce from had as input n arithmetic circuits capturing the cumulative valuation of n agents on $[0,1]$, which were piece-wise polynomials of maximum degree 2. However, since their (density) valuation functions consist of only rectangles and triangles, the input can also be sets of points that define the aforementioned shapes. So, there is no need for extra translation of the input of Consensus-Halving to the input of SC-Pizza-Sharing.

FIXP-hardness. Here we show that finding an exact solution to SC-Pizza-Sharing is FIXP-hard. The key difference between this reduction and the previous reductions on the approximate versions is that the instance I_{CH}^{DFMS} contains triangular shaped valuations for agents. More specifically, all of the following hold:

1. the valuation function of every agent is 4-block+triangle, or 6-block;
2. every triangle has height 2 and belongs to exactly one interval of interest of the form $[a, a+1]$;

3. for every agent $i \in [n]$ there exists an interval $[a, b]$ that contains more than half of their total valuation and in addition for every $i \neq i'$ we have $(a, b) \cap (a', b') = \emptyset$.

We will follow the same approach as in the overlapping reduction. Namely, every interval of interest will correspond to a unit-square located on the diagonal of the instance. Block-shaped valuations will be translated as before. For the triangular valuations, Point 2 guarantees that no triangle is split over two different intervals of interest. Hence, we can create an axis-aligned triangle inside the unit-square. Formally, if there exists a triangle in the interval of interest $[x_j, x_{j+1}]$, then we create a triangle with vertices $(j, j+1)$, $(j+1, j)$, $(j+1, j+1)$ with density 1; this is because all triangles in the Consensus-Halving instance have the same value. Figure 8 shows an example of this.

One complication is that, if the SC-path turns inside a square, then it may not be possible to map this back to cuts in the SC-Pizza-Sharing instance. One example of this would be when the path cuts a portion of the square without cutting the triangle at all. Fortunately, we are able to use point 3 above to show that all exact solutions to the SC-Pizza-Sharing instance must pass through n distinct squares. Therefore, if a SC-path "wastes" a turn by turning inside a square, then it can pass through at most $n-1$ distinct squares, and so it cannot be an exact solution to the SC-Pizza-Sharing instance.

Hence we can restrict ourselves to SC-paths that do not turn inside a square. The mapping from SC-paths to Consensus-Halving cuts is the same as the overlapping reduction. Here, in particular, we rely on the fact that if the SC-path does not turn inside a square, then it must pass through the triangular mass either horizontally or vertically, and so we can map this back to a single cut in the Consensus-Halving instance.

The following lemma is proved formally in Appendix I.

Lemma 15. *Any* SC*-path that is a solution for the produced* SC-Pizza-Sharing *instance, does not turn inside any unit-square.*

Combining Lemma 15 and the FIXP-hardness proven for I_{CH}^{DFMS} in [7] gives us the following result.

Theorem 16. SC-Pizza-Sharing *is* FIXP*-hard even when every mass distribution consists of at most six pieces that can be unit-squares or right-angled triangles, and they have overlap 3.*

ETR-hardness. We can also show that deciding whether there is an exact SC-Pizza-Sharing solution with $n-2$ turns exists is ETR-hard. To show this, we will use a result of [7], where it was shown that deciding whether there exists an exact Consensus-Halving solution with n agents and $n-1$ cuts is ETR-hard. We give a reduction from this version of Consensus-Halving to the decision problem for SC-Pizza-Sharing. The reduction uses the same ideas that we presented for the BU-hardness reduction. The full details are in Appendix J, where the following theorem is shown.

Theorem 17. *It is* ETR*-hard to decide if an exact* SC-Pizza-Sharing *instance admits a solution with a* SC*-path with* $n-2$ *turns.*

5 Conclusion

We studied the complexity of SC-Pizza-Sharing and we have shown that finding an approximate solution is PPA-complete even in very simple scenarios. One advantage of our reductions is that they are quite general and can be used as black boxes to get further results. For example, if we can prove that ε-Consensus-Halving is PPA-hard for a *constant* ε and block valuations, the Overlapping reduction implies PPA-hardness for ε-SC-Pizza-Sharing.

Our work identifies several interesting problems that require extra investigation. The most important one is whether we can get in polynomial time a solution for ε-SC-Pizza-Sharing for some *constant* ε. What is the exact complexity SC-Pizza-Sharing when every mass distribution is a collection of axis-aligned rectangles? Our results show that the problem is PPA-hard and that it belongs to BU. Is it complete for any of these classes? Another problem is the complexity of ε-SC-Pizza-Sharing when every mass distribution consists of a constant number of non-overlapping rectangles. Also, we have shown that computing an exact solution of SC-Pizza-Sharing is FIXP-hard and in BU. Is the problem complete for FIXP or BU? We conjecture that it is BU-complete.

Acknowledgements

The work of the third author has been supported by the Alexander von Humboldt Foundation with funds from the German Federal Ministry of Education and Research (BMBF).

References

[1] Pankaj K Agarwal, Jeff Erickson, et al. Geometric range searching and its relatives. *Contemporary Mathematics*, 223:1–56, 1999.

[2] Takao Asano, Tetsuo Asano, and Ron Y. Pinter. Polygon triangulation: Efficiency and minimality. *J. Algorithms*, 7(2):221–231, 1986.

[3] Luis Barba, Alexander Pilz, and Patrick Schnider. Sharing a pizza: bisecting masses with two cuts. *arXiv preprint arXiv:1904.02502*, 2019.

[4] Pavle Blagojević, Florian Frick, Albert Haase, and Günter Ziegler. Topology of the Grünbaum–Hadwiger–Ramos hyperplane mass partition problem. *Transactions of the American Mathematical Society*, 370(10):6795–6824, 2018.

[5] John Canny. Some algebraic and geometric computations in PSPACE. In *Proc. of STOC*, pages 460–467, New York, NY, USA, 1988. ACM.

[6] Jesús De Loera, Xavier Goaoc, Frédéric Meunier, and Nabil Mustafa. The discrete yet ubiquitous theorems of Carathéodory, Helly, Sperner, Tucker, and Tverberg. *Bulletin of the American Mathematical Society*, 56(3):415–511, 2019.

[7] Argyrios Deligkas, John Fearnley, Themistoklis Melissourgos, and Paul G. Spirakis. Computing exact solutions of consensus halving and the Borsuk-Ulam theorem. *Journal of Computer and System Sciences*, 117:75 – 98, 2021.

[8] Argyrios Deligkas, Aris Filos-Ratsikas, and Alexandros Hollender. Two's company, three's a crowd: Consensus-halving for a constant number of agents. *CoRR*, abs/2007.15125, 2020.

[9] Herbert Edelsbrunner. *Algorithms in combinatorial geometry*, volume 10. Springer Science & Business Media, 2012.

[10] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.

[11] Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.

[12] Aris Filos-Ratsikas, Søren Kristoffer Still Frederiksen, Paul W. Goldberg, and Jie Zhang. Hardness Results for Consensus-Halving. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 24:1–24:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.

[13] Aris Filos-Ratsikas and Paul W. Goldberg. Consensus Halving is PPA-complete. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 51–64. ACM, 2018.

[14] Aris Filos-Ratsikas and Paul W. Goldberg. The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, pages 638–649. ACM, 2019.

[15] Aris Filos-Ratsikas, Alexandros Hollender, Katerina Sotiraki, and Manolis Zampetakis. Consensus-Halving: Does it Ever Get Easier? In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 381–399, 2020.

[16] Aris Filos-Ratsikas, Alexandros Hollender, Katerina Sotiraki, and Manolis Zampetakis. A topological characterization of modulo-p arguments and implications for necklace splitting. *arXiv preprint*, 2020.

[17] M. R. Garey, David S. Johnson, Franco P. Preparata, and Robert Endre Tarjan. Triangulating a simple polygon. *Inf. Process. Lett.*, 7(4):175–179, 1978.

[18] Subir Kumar Ghosh and David M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20(5):888–910, 1991.

[19] Atsushi Kaneko and Mikio Kano. Discrete geometry on red and blue points in the plane—a survey—. In *Discrete and computational geometry*, pages 551–570. Springer, 2003.

[20] Mikio Kano and Jorge Urrutia. Discrete geometry on red and blue points in the plane-a survey. *Graphs and Combinatorics*, 2020.

[21] Roman N Karasev, Edgardo Roldán-Pensado, and Pablo Soberón. Measure partitions using hyperplanes with fixed directions. *Israel Journal of Mathematics*, 212(2):705–728, 2016.

[22] Jiri Matousek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.

[23] Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. *CoRR*, abs/1406.2636, 2014.

[24] Jiří Matoušek. *Using the Borsuk-Ulam theorem: lectures on topological methods in combinatorics and geometry*. Springer Science & Business Media, 2008.

[25] Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

[26] Kurt Mehlhorn. *Data Structures and Algorithms 3: Multi-Dimensional Searching and Computational Geometry*. Springer-Verlag, Berlin, Heidelberg, 1984.

[27] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[28] Edgardo Roldán-Pensado and Pablo Soberón. A survey of mass partitions. *CoRR*, abs/2010.00478, 2020.

[29] Karthik C. S. and Arpan Saha. Ham sandwich is equivalent to Borsuk-Ulam. In Boris Aronov and Matthew J. Katz, editors, *33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[30] Marcus Schaefer. Complexity of some geometric and topological problems. In *International Symposium on Graph Drawing*, pages 334–344. Springer, 2009.

[31] Forest W. Simmons and Francis E. Su. Consensus-halving via theorems of Borsuk-Ulam and Tucker. *Mathematical social sciences*, 45(1):15–25, 2003.

[32] Rade Živaljević. Topological methods in discrete geometry. In *Handbook of Discrete and Computational Geometry, Third Edition*. Taylor & Francis, 2017.

A Input representation of SC-Pizza-Sharing.

While the mathematical proof of Theorem 3 holds for general measures which are absolutely continuous with respect to the Lebesgue measure, in the computational problem SC-Pizza-Sharing we need a standardized way to describe the input, and therefore restrict to particular classes of measures. We consider the class of mass distributions that are defined by weighted polygons with holes. This class consists of mass distributions with the property that are succinctly represented in the input of a Turing machine.

We will use the standard representation of 2-d polygons in computational geometry problems, that is, a directed chain of points. Consider a polygon that is defined by k points $p_i = (x_i, y_i)$, where $x_i, y_i \in [0,1] \cap \mathbb{Q}$, for $i \in [k]$, which form a directed chain $C = (p_1, \ldots, p_k)$. This chain represents a closed boundary defined by the line segments (p_i, p_{i+1}) for $i \in [k-1]$ and a final one (p_k, p_1). Since we consider polygons with holes, we need a way to distinguish between the polygons that define a boundary whose interior has strictly positive weight and polygons that define the boundary of the holes (whose interior has zero weight). We will call the former *solid* and the latter *hollow* polygon. To distinguish between the two, we define a solid polygon to be represented by directed line segments with counterclockwise orientation, while a hollow polygon to be represented similarly but with clockwise orientation. Furthermore, each solid polygon C_s, its weight w and its $r \geq 0$ holes $C_{h_1}, C_{h_2}, \ldots, C_{h_r}$ in the interior, are grouped together in the input to indicate that all these directed chains of points represent a single polygon $(w, C_s, C_{h_1}, \ldots, C_{h_r})$.

B Proof of Theorem 3

Consider the exact SC-Pizza-Sharing problem, whose definition is the same as Definition 2 for $\varepsilon = 0$. We are given n measures in $\mathbb{R}^2$. For ease of presentation, we consider the measures to be normalized in $[0,1]^2$ instead of $\mathbb{R}^2$.

We shall first consider the case where n is even, i.e. $n = 2m$ for some $m \in \mathbb{N} \setminus \{0\}$. The proof for $n = 2m-1$ is similar and is mentioned at the end of this case. Let us first split the unit-square into $m+1 \leq n$ *stripes* using m horizontal cuts $0 \leq y_1 \leq y_2 \leq \cdots \leq y_m \leq 1$, and denote also by $y_0 = 0$ and $y_{m+1} = 1$. Then, let us cut with directed (pointing upwards) vertical lines $x_1, x_2, \ldots, x_m \in [0,1]$ each of the stripes, except for the bottom one, where a cut x_i starts from the horizontal segment $y = y_i$ and ends up touching $y = y_{i+1}$. As stated earlier, the bottom stripe $[0, y_1]$ is not cut by any vertical cut (see Figure 3 for an example).

We also define the variables $z_1, z_2, \ldots, z_{m+1}$ where $|z_i| = y_i - y_{i-1}$, $i \in [m+1]$. The sign of z_i, $i \in [m+1]$ indicates the sign of the leftmost part of slice $[y_{i-1}, y_i]$ that x_{i-1} defines, and we set the whole slice $[0, y_1]$ to have the sign of z_1. Clearly, $(z_1, z_2, \ldots, z_{m+1}) \in S^m$, since $\sum_{i=1}^{m+1} |z_i| = 1$, and $(x_1, \ldots, x_m) \in [0,1]^m$, by definition. A *feasible solution* of SC-Pizza-Sharing is then the vector $(z_1, \ldots, z_{m+1}, x_1, \ldots, x_m)$; this defines a directed path that consists of horizontal and vertical line segments with at most $2m-1$ turns. We can recover this path using Algorithm 1. Note that in the algorithm we implicitly suggest to navigate using the following trick: if we are at a horizontal part of the path, say $y = y'$, and we hit the boundary $x = 0$ or $x = 1$ before we reach a vertical cut, then we wrap around in the horizontal dimension and continue from point $(1, y')$ or $(0, y')$ respectively.

We will now define a Borsuk-Ulam function, namely a continuous function $f : S^d \mapsto \mathbb{R}^d$, for a suitable dimension $d > 0$ to be defined later. It will turn out that $d = 2m = n$, but having it undetermined for as long as we can makes the proof transparent enough to help in the understanding of the cases where $d \neq n$ (see Theorems 6, 9 and their proofs). Let the variables $Z_1, \ldots, Z_m, R, X_1, \ldots, X_m$ be in the $(2m)$-sphere S^{2m} under the L_1 norm, i.e. $\sum_{i=1}^{m} |Z_i| + |R| + \sum_{i=1}^{m} |X_i| = 2m$. Notice that now instead of a Z_{m+1} variable we have R. A vector in S^{2m} maps back to a feasible solution $(z_1, \ldots, z_{m+1}, x_1, \ldots, x_m)$ of SC-Pizza-Sharing in the following way:

$$|z_i| = \min\left\{\sum_{j=1}^{i} |Z_j|, 1\right\} - \sum_{j=1}^{i-1} |z_j| = \begin{cases} |Z_i| & \text{, if} \quad \sum_{j=1}^{i} |Z_j| < 1, \\ 1 - \sum_{j=1}^{i-1} |z_j| & \text{, otherwise,} \end{cases}$$

$$|z_{m+1}| = 1 - \sum_{i=1}^{m} |z_j|,$$

$$\text{and} \quad \text{sign}(z_i) = \text{sign}(Z_i), \quad \forall i \in [m]$$
$$\text{sign}(z_{m+1}) = \text{sign}(R).$$

Also,

$$x_i = \min\{|X_i|, 1\}, \quad \text{and} \quad \text{sign}(x_i) = \text{sign}(X_i), \quad \forall i \in [m].$$

Finally, we highlight that in the map-back process, R is only used to indicate the sign of variable z_{m+1}, while its absolute value has no other purpose than to serve as a *remainder*: it ensures that $(Z_1, \ldots, Z_m, R, X_1, \ldots, X_m) \in S^{2m}$, i.e. it is $|R| = 2m - (\sum_{i=1}^{m} |Z_i| + \sum_{i=1}^{m} |X_i|)$.

For any given point $\vec{P} = (Z_1, \ldots, Z_m, R, X_1, \ldots, X_m) \in S^{2m}$, the Borsuk-Ulam function is defined to be the total "+" (positive) measure on $[0,1]^2$ induced by $\vec{P}$, and we denote it by $\mu(R^+; \vec{P})$, that is, $f(\vec{P}) = \mu(R^+; \vec{P})$. The total positive measure is a continuous function of the variables. f is mapped on $\mathbb{R}^{2m}$ (i.e. $d = 2m = n$), therefore it affords to capture $2m = n$ distinct measures. By the Borsuk-Ulam theorem, there exist two antipodal points $\overrightarrow{P^*}, -\overrightarrow{P^*} \in S^{2m}$ such that $f(\overrightarrow{P^*}) = f(-\overrightarrow{P^*})$. Notice that $f(-\vec{P}) = \mu(R^-; \vec{P})$, since by flipping the signs of the variables of $\vec{P}$, we consider the "$-$" (negative) measure of $[0,1]^2$ induced by $\vec{P}$. Therefore, when $f(\overrightarrow{P^*}) = f(-\overrightarrow{P^*})$ we will have $\mu(R^+; \overrightarrow{P^*}) = \mu(R^-; \overrightarrow{P^*})$, that is, in each of the $2m$ measures, the positive total measure equals the negative one. By mapping back $\overrightarrow{P^*} \in S^{2m}$ to the corresponding point $\vec{p} = (z_1, \ldots, z_{m+1}, x_1, \ldots, x_m) \in S^m \times [0,1]^m$, we get a solution to SC-Pizza-Sharing. The total number of turns of the directed path is $2m - 1 = n - 1$.

The proof is similar when $n = 2m - 1$. The horizontal cuts are again $0 \leq y_1 \leq y_2 \leq \cdots \leq y_m \leq 1$, but the vertical cuts are $x_1, x_2, \ldots, x_{m-1} \in [0,1]$, meaning that the top slice is not vertically cut. Also, it is easy to see that one could consider the path to be again y-monotone but in the opposite direction, meaning that there is no line segment pointing upwards.

Remark 18. *Note that a symmetric proof exists, where the slices are vertical instead of horizontal, and the cuts within the slices are horizontal instead of vertical. The analysis is similar to the one we give here, and it guarantees the existence of an* SC*-path which is allowed to wrap around in the vertical dimension, it bisects all n measures and is x-monotone with no line segment heading left (or right).*

Algorithm 1 Mapping labelled cuts to SC-paths

Input: A vector $(z_1, \ldots, z_{k+1}, x_1, \ldots, x_k)$.
Output: A feasible solution to SC-Pizza-Sharing (i.e. a SC-path).

1: Find the set $T = \{t_1 \ldots, t_m\}$ of $m \leq k+1$ of all indices of $z_1, \ldots, z_{k+1}$, where $t_1 < \cdots < t_m$ and for each $\ell \in [m]$ it holds that $z_{t_\ell} \neq 0$.
2: $i \leftarrow 1$
3: $i' \leftarrow 2$
4: **while** $i' \leq m$ **do**
5: **if** $z_{t_i} \cdot z_{t_{i'}} \cdot (x_{t_{i'-1}} - x_{t_{i-1}}) > 0$ **then**
6: Go **right** ($\rightarrow$)
7: **if** $z_{t_i} \cdot z_{t_{i'}} \cdot (x_{t_{i'-1}} - x_{t_{i-1}}) < 0$ **then**
8: Go **left** ($\leftarrow$)
9: **if** $z_{t_i} \cdot z_{t_{i'}} > 0$ and $x_{t_{i'-1}} - x_{t_{i-1}} = 0$ **then**
10: Go **up** ($\uparrow$)
11: **if** $z_{t_i} \cdot z_{t_{i'}} < 0$ and $x_{t_{i'-1}} - x_{t_{i-1}} = 0$ **then**
12: Go **left** ($\leftarrow$)
13: $i \leftarrow i + 1$
14: $i' \leftarrow i' + 1$

C Proof of Theorem 5

Consider an arbitrary instance of exact SC-Pizza-Sharing, i.e. the one of Definition 2, where we additionally restrict the mass distributions to be weighted polygons (with holes). For ease of presentation, we will give distinct colours to the mass distributions μ_i, $i \in [n]$ and call them *colours* $1, 2, \ldots, n$. We are given a set of polygons with holes in the input form described in Appendix A. We will first do a preprocessing of the input: (a) normalization in $[0,1]^2$, and (b) triangulation. The former is needed in order to simulate the space of the mathematical existence proof of Theorem 3, while the latter allows us to construct the Borsuk-Ulam function of the aforementioned proof via circuits. For the computation of the Borsuk-Ulam function described in Appendix B we need a way of computing the "positive" measure of a polygon, as dictated by a given feasible solution $\vec{P}$. To simplify this computation, we will triangulate further the polygon to end up with only non-obtuse triangles, which can be decomposed into axis-aligned right-angled triangles (Appendix C.1).

For task (a) we first check among all polygons of all colours, what the smallest and greatest coordinates of their vertices are, which defines a rectangle that includes all our polygons. Then, we find the square with smallest sides within which the aforementioned rectangle can be inscribed. Finally, by shifting the rectangle so that its bottom left vertex is at $(0,0)$, and then scaling it down so that each side is of length 1, we get a normalized input where each polygon is in $[0,1]^2$ and their relative positions and areas are preserved. Task (b) is easily taken care of via already known algorithms (e.g. [17, 2, 26, 18]) that triangulate polygons with holes in $O(n \log n)$ time (which is optimal) without inserting additional vertices.

C.1 Computing areas of polygons via axis-aligned right-angled triangle decomposition

Here we first show how an arbitrary triangle can be decomposed into right-angled triangles whose right angle is additionally axis-aligned. Then, by using only the allowed gates of BU, we present a way to construct a Borsuk-Ulam function. We show that any solution of Borsuk-Ulam whose input is the aforementioned function can be mapped back in polynomial time to an exact SC-Pizza-Sharing solution.

By the definition of the Borsuk-Ulam function of Section 3, it is apparent that we need to be able to compute parts of the area of a polygon, depending on where square-cuts fall. The function we provide can compute parts of the area of axis-aligned right-angled triangles. For that reason, after a standard triangulation (e.g. using the technique of [2]) we further preprocess it and decompose each obtuse triangle into two right-angled triangles, by adding an extra line segment.

In particular, we check the obtuseness of a triangle $\widehat{ABC}$ by computing the squared lengths of its sides AB^2, BC^2, AC^2 (each is rational; a sum of squares of rationals), taking the largest one, w.l.o.g. AC^2 and then checking whether $AB^2 + BC^2 < AC^2$. If the inequality is not true then $\widehat{ABC}$ is non-obtuse and we proceed. Otherwise, we add the line segment BD that starts from B and ends at D on side AC, where $\widehat{BDC} = \widehat{ADB} = 90^\circ$. The coordinates (x_D, y_D) of D are rationals since they are the solution of the following two equations: (a) one that dictates that D is on AC: $\frac{y_D - y_A}{x_D - x_A} = \frac{y_A - y_C}{x_A - x_C}$, and (b) one that captures the fact that BD and AC are perpendicular: $\frac{y_B - y_D}{x_B - x_D} \cdot \frac{y_A - y_C}{x_A - x_C} = -1$. In fact,

$$x_D = \frac{(y_A - y_D)[(y_B - y_A)(x_A - x_C) + (y_A - y_C)x_A] + x_B(x_A - x_C)^2}{(x_A - x_C)(x_A - x_C + y_A - y_C)}, \quad \text{and}$$

$$y_D = \frac{y_A - y_C}{x_A - x_C}(x_D - x_A) + y_A.$$

At this point the triangulation of each polygon consists of non-obtuse triangles. The following proposition makes the computation of areas of these triangles' parts possible via a decomposition into axis-aligned right-angled triangles.

Proposition 19. *The area of any non-obtuse triangle can be computed by the areas of five axis-aligned right-angled triangles.*

Proof. A proof by picture is presented in Figure 4, where we draw a segment from the top-left corner to the bottom-right one, and $\widehat{XYB} + \widehat{XBZ} - \widehat{AYB} - \widehat{XAC} - \widehat{CBZ}$.

The proof is immediate if we show that every non-obtuse triangle $\widehat{ABC}$ can be *tightly inscribed* inside a rectangle, meaning that all of its vertices touch the rectangle's perimeter. In particular, the rectangle has coordinates $(\min\{x_A, x_B, x_C\}, \min\{y_A, y_B, y_C\})$, $(\max\{x_A, x_B, x_C\}, \min\{y_A, y_B, y_C\})$, $(\max\{x_A, x_B, x_C\}, \max\{y_A, y_B, y_C\})$, and $(\min\{x_A, x_B, x_C\}, \max\{y_A, y_B, y_C\})$.

First, we argue that one of $\widehat{ABC}$'s vertices is on a corner of the rectangle. W.l.o.g. suppose $\arg\min\{x_A, x_B, x_C\} = A$. If $\arg\min\{y_A, y_B, y_C\}) = A$ or $\arg\max\{y_A, y_B, y_C\}) = A$ then A is the bottom-left or top-left corner of the rectangle, respectively. Otherwise, suppose w.l.o.g. that $\arg\min\{y_A, y_B, y_C\}) = B$. If $\arg\max\{x_A, x_B, x_C\} = B$ then B is the bottom-right corner of the rectangle. Otherwise, $\arg\max\{x_A, x_B, x_C\} = C$. Then, if $\arg\max\{y_A, y_B, y_C\}) = C$, C is the top-right corner of the rectangle. Otherwise, $\arg\max\{y_A, y_B, y_C\}) = A$ and A is the top-left corner of the rectangle. We conclude that a vertex of the triangle, w.l.o.g. A, is on a corner of the rectangle.

Now we need to show that B and C lie on the perimeter of the rectangle. By the definition of the aforementioned rectangle's vertices, it cannot be that both C and D are not on the perimeter; if for example A is the top-right corner then $\arg\min\{y_A, y_B, y_C\} \in \{B, C\}$, therefore vertex $\arg\min\{y_A, y_B, y_C\}$ touches the lower side of the rectangle, and similarly if A is any of the other corners. Now, for the sake of contradiction, suppose that the remaining vertex, w.l.o.g. C does not touch the perimeter of the rectangle. Then, by definition of the rectangle's vertices, B is on another corner. If A and B are on the same side of the rectangle, then it is clear by the definition of the rectangle's coordinates that C must be on the perimeter of it. Otherwise A and B are diagonal corners of the rectangle. Then, AB is the largest side of the triangle and if C is not on the boundary, it holds that $AC^2 + BC^2 < AB^2$, meaning that $\widehat{ABC}$ is obtuse, a contradiction. Therefore, any non-obtuse triangle can be tightly inscribed in a rectangle. □

C.2 Constructing the Borsuk-Ulam function

Here we show how to construct the Borsuk-Ulam function $f : S^n \mapsto \mathbb{R}^n$ given n sets of weighted polygons. We will focus on an arbitrary colour $i \in [n]$ and present the coordinate f_i.

Consider the $\tau \geq 1$ weighted polygons of the i-th colour, and let us focus on a particular polygon $t \in [\tau]$. We have triangulated the polygon into m_t non-obtuse triangles. Consider one such triangle $j \in [m_t]$ and the *virtual triangles* $T_j^1, T_j^2, T_j^3, T_j^4, T_j^5$, which are the corresponding five axis-aligned right-angled triangles described in Appendix C.1. W.l.o.g. we consider T_j^1 and T_j^2 to be the *positively contributing triangles* and the rest being the *negatively contributing triangles*. For each of them we will be computing the positive measure that the cut $\vec{p}$ of SC-Pizza-Sharing defines (see Appendix B). By the axis-aligned right-angled triangle decomposition described in the proof of Proposition 19, it suffices to show how to compute parts of areas of such a triangle, for all of its four possible *orientations*: $Q_I, Q_{II}, Q_{III}, Q_{IV}$, where Q_o is the orientation when, by shifting the triangle so that the vertex of the right angle is on $(0, 0)$, the whole triangle is in the o-th quadrant.

First, we test the orientation of our triangle. Observe that for T_j^1 and T_j^2 we know that the orientation is Q_I and Q_{III}, respectively (see $\widehat{XYB}$ and $\widehat{XZB}$ in Figure 4). For T_j^3, T_j^4, T_j^5, we check the coordinates and identify what kind of triangles they are.

For a fixed colour $i \in [n]$, for each possible category Q_o, $o \in \{I, II, III, IV\}$ we show how to compute the term that an axis-aligned right-angled triangle $T_j^r = \widehat{ABC}$, $j \in [m]$, $r \in [5]$ contributes to the Borsuk-Ulam function $f(\vec{p})_i$. Let us denote by $g_{j,r}(x)$ the linear function of the line segment of $\widehat{ABC}$'s hypotenuse (with respect to the horizontal axis x). We also denote by $(x_{j,r}, y_{j,r})$ and $(x'_{j,r}, y'_{j,r})$ the bottom and top points that define the hypotenuse of $\widehat{ABC}$. The vertex of the right angle is defined by these four coordinates depending on Q_o.

For any given point (feasible solution) $\vec{p}$ that defines a SC-path of SC-Pizza-Sharing (see Appendix B), recall that there are $\lceil n/2 \rceil$ slices $0 = y_0 \leq y_1 \leq \cdots \leq y_{\lceil n/2 \rceil} \leq y_{\lceil n/2 \rceil + 1} = 1$ that determine $\lceil n/2 \rceil + 1$ slices $|z_1|, \ldots, |z_{\lceil n/2 \rceil + 1}|$ that partition $[0, 1]^2$. One can see that, given the z_s's, $s \in [\lceil n/2 + 1 \rceil]$ from $\vec{p}$, the y_s's can be computed by the recursive expression $y_s = |z_s| - y_{s-1}$.

We first define the following auxiliary functions $A_j^r(z_s)$, $B_j^r(-z_s)$ for each Q_o:

$\mathbf{Q_I}$:

$$\begin{aligned}
A_j^r(z_s) &:= \frac{1}{2}\cdot[\min\{\max\{x_{j,r}-x_{s-1},0\}, g_{j,r}^{-1}(y_{s-1})-x'_{j,r}\}+\\
&\quad+\min\{\max\{g_{j,r}^{-1}\left(y_{s-1}+\max\{-z_s,0\}\right)-x_{s-1},0\}, g_{j,r}^{-1}\left(y_{s-1}+\max\{-z_s,0\}\right)-x'_{j,r}\}]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y_{s-1}-y_{j,r},0\}+\max\{-z_s,0\}\}.\\
B_j^r(-z_s) &:= \frac{1}{2}\cdot\left[(x'_{j,r}-x_{j,r})+\max\{x'_{j,r}-g_{j,r}^{-1}\left(y_{s-1}+\max\{-z_s,0\}\right),0\}\right]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y_{s-1}-y_{j,r},0\}+\max\{-z_s,0\}\}.
\end{aligned}$$

$\mathbf{Q_{II}}$:

$$\begin{aligned}
A_j^r(z_s) &:= \frac{1}{2}\cdot[\min\{\max\{x_{s-1}-x_{j,r},0\}, x'_{j,r}-g_{j,r}^{-1}(y_{s-1})\}+\\
&\quad+\min\{\max\{x_{s-1}-g_{j,r}^{-1}\left(y_{s-1}+\max\{z_s,0\}\right),0\}, x'_{j,r}-g_{j,r}^{-1}\left(y_{s-1}+\max\{z_s,0\}\right)\}]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y_{s-1}-y_{j,r},0\}+\max\{z_s,0\}\}.\\
B_j^r(-z_s) &:= \frac{1}{2}\cdot\left[(x'_{j,r}-x_{j,r})+\max\{x'_{j,r}-g_{j,r}^{-1}\left(y_{s-1}+\max\{-z_s,0\}\right),0\}\right]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y_{s-1}-y_{j,r},0\}+\max\{-z_s,0\}\}.
\end{aligned}$$

$\mathbf{Q_{III}}$:

$$\begin{aligned}
A_j^r(z_s) &:= \frac{1}{2}\cdot[\min\{\max\{x_{s-1}-x'_{j,r},0\}, x_{j,r}-g_{j,r}^{-1}(1-y_{s-1})\}+\\
&\quad+\min\{\max\{x_{s-1}-g_{j,r}^{-1}\left((1-y_{s-1})+\max\{z_s,0\}\right),0\}, x_{j,r}-g_{j,r}^{-1}\left((1-y_{s-1})+\max\{z_s,0\}\right)\}]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y'_{j,r}-y_{s-1},0\}+\max\{z_s,0\}\}.\\
B_j^r(-z_s) &:= \frac{1}{2}\cdot\left[(x_{j,r}-x'_{j,r})+\max\{x_{j,r}-g_{j,r}^{-1}\left(y_{s-1}+\max\{-z_s,0\}\right),0\}\right]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y'_{j,r}-y_{s-1},0\}+\max\{-z_s,0\}\}.
\end{aligned}$$

$\mathbf{Q_{IV}}$:

$$\begin{aligned}
A_j^r(z_s) &:= \frac{1}{2}\cdot[\min\{\max\{x'_{j,r}-x_{s-1},0\}, g_{j,r}^{-1}(1-y_{s-1})-x_{j,r}\}+\\
&\quad+\min\{\max\{g_{j,r}^{-1}\left((1-y_{s-1})-x_{s-1}+\max\{-z_s,0\}\right),0\}, g_{j,r}^{-1}\left((1-y_{s-1})-x_{j,r}+\max\{-z_s,0\}\right)\}]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y'_{j,r}-y_{s-1},0\}+\max\{-z_s,0\}\}.\\
B_j^r(-z_s) &:= \frac{1}{2}\cdot\left[(x'_{j,r}-x_{j,r})+\max\{g_{j,r}^{-1}\left(y_{s-1}+\max\{z_s,0\}\right)-x_{j,r},0\}\right]\cdot\\
&\quad\cdot\min\{y'_{j,r}-y_{j,r}, \max\{y'_{j,r}-y_{s-1},0\}+\max\{z_s,0\}\}.
\end{aligned}$$

For an example of case Q_{II} see Figure 9.

Then, we get the total area that *all slices* up to z_s (i.e. $[0, y_s]$) define together with x_s from

$$D_j^r(s,z_s) := A_j^r(z_s)+\left(B_j^r(-z_s)-A_j^r(-z_s)\right),$$

which has the property that if $z_s > 0$ (resp $z_s < 0$), then the computed area considers the thickness $|z_s|$ only in the part of the slice that is left (resp. right) to x_s. For $z_s = 0$, $|z_s|$ has no thickness, its measure is 0, and consistently vanishes from the above functions.

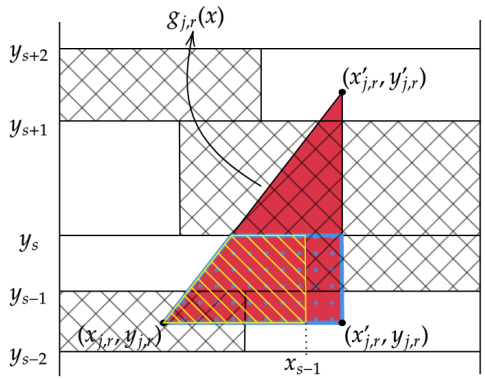


Figure 9: An example of a Q_{II} triangle. The yellow (diagonally shaded) trapezoid is computed by function $A^r_j(z_s)$ and the blue (dotted) trapezoid is computed by $B^r_j(-z_s)$.

Then, to isolate the part that *only slice* z_s contributes to the positive measure, we define

$$d^r_j(s) := D^r_j(s, z_s) - D^r_j(s, 0).$$

Consequently, the positive measure that an (unweighted) non-obtuse triangle j contributes to the Borsuk-Ulam function according to the SC-path $\vec{p}$ is

$$q_j := \sum_{s=1}^{\lceil n/2+1 \rceil} \left(d^1_j(s) + d^2_j(s) - d^3_j(s) - d^4_j(s) - d^5_j(s) \right).$$

Finally, given that colour $i \in [n]$ has τ many weighted polygons, each of weight w_t, $t \in [\tau]$, which has been decomposed into m_t many non-obtuse triangles, i's positive measure (i.e. the i-th coordinate of the Borsuk-Ulam function) is

$$f(\vec{p})_i = \sum_{t=1}^{\tau} w_t \sum_{j=1}^{m_t} q_j.$$

D Proof of Theorem 6

The proof is by showing that the aforementioned problem can be formulated as an ETR problem. We will use the machinery of the BU containment proof (Theorem 5) to build the piece-wise polynomial function f. In the proof of the aforementioned theorem we construct the function $f : S^n \mapsto R^n$ so that it computes the "positive" part of measure $i \in [n]$ in its i-th coordinate, and we require (at most) $n-1$ turns in the SC-path. It is easy to modify this construction for any given number k of turns: by the proof of Theorem 3, what we need to do is to cut $\lceil (k+1)/2 \rceil + 1$ slices (i.e. place $\lceil (k+1)/2 \rceil$ horizontal cuts) in $[0,1]^2$ and another $k+1-\lceil (k+1)/2 \rceil$ vertical cuts on them starting with the second slice from the bottom. Then, by the same mapping as in the aforementioned proof, we have a function $f : S^{k+1} \mapsto R^n$ for which, if $f(\vec{P}^*) = f(-\vec{P}^*)$ for some $\vec{P}^*$ we have a solution to exact SC-Pizza-Sharing with k turns in its SC-path. One can see that the SC-Pizza-Sharing problems of Theorems 3 and 5 were special cases for $k \geq n-1$, for which the aforementioned equality is guaranteed by the Borsuk-Ulam theorem.

The requirement $f(\vec{P}) = f(-\vec{P})$ can be implemented as an ETR formula, that is, it can be written in the form: $\exists \vec{P} \in \mathbb{R}^m \cdot \Phi$, where Φ is a boolean formula using connectives $\{\wedge, \vee, \neg\}$ over polynomials with domain $\mathbb{R}^m$ for some $m \in \mathbb{N}$ compared with the operators $\{<, \leq, =, \geq, >\}$. Recall from Appendix C.2 that function f is constructed using the operations $\{c, +, -, \times c, \times, \max, \min\}$, since also the $|\cdot|$ operation can be implemented as: $|t| := \max\{t, 0\} + \max\{-t, 0\}$. In f, we only need to replace each $\max\{y, z\}$ and $\min\{y, z\}$ occurrence with a new variable $g_{\max}$ and $g_{\min}$ respectively and include the

following constraints C_{max}, C_{min} in formula Φ:

$$\begin{aligned} \text{for max:} \quad C_{max} &= ((g_{\max} = y) \wedge (y \geq z)) \vee ((g_{\max} = z) \wedge (z > y)), \\ \text{for min:} \quad C_{min} &= ((g_{\min} = y) \wedge (y \leq z)) \vee ((g_{\min} = z) \wedge (z < y)). \end{aligned}$$

Let us denote by $\vec{g} \in \mathbb{R}^{m'}$ all the variables needed to replace the $m' \in \mathbb{N}$ occurrences of max and min operations in f. Also, denote by C the conjunction of all the aforementioned C_{max}, C_{min} constraints. Then our ETR formula is:

$$\exists \vec{P}, \vec{g} \in \mathbb{R}^{m+m'} \cdot \left(\sum_{j=1}^{k+2} |P_j| = k+1 \right) \wedge C \wedge \left(\bigwedge_{i=1}^{n} f(\vec{P})_i = f(-\vec{P})_i \right).$$

This completes the proof.

E Proof of Theorem 8

The proof is straight-forward with the help of the proof of Theorem 5. Given the ε-SC-Pizza-Sharing instance with n mass distributions, we just have to construct the Borsuk-Ulam function f with a circuit as in the aforementioned proof (a task implementable in polynomial time). The function, as the proof shows, is continuous piece-wise polynomial, and therefore it is λ-Lipschitz continuous for $\lambda = \max_{j=1}^{n+1} \left\{ \sup_x \left|\left| \frac{\partial f(x)}{\partial x_j} \right|\right|_\infty \right\}$ (and note that points where $f_i(x)$, $i \in [n]$ is non-differentiable do not matter for Lipschitzness). By the definition of f (proof of Theorem 5) one can see that λ is constant: the polynomial pieces of f are of degree at most 2, and the partial derivative of each f_i is defined by the rational points given in the input that define the polygons. By formulating it as an ε-Borsuk-Ulam instance, any solution of that instance corresponds to a solution of ε-SC-Pizza-Sharing, since $f_i(x)$ and $f_i(-x)$ capture the measures $\mu_i(\mathbb{R}^+)$ and $\mu_i(\mathbb{R}^-)$, respectively in the definition of ε-SC-Pizza-Sharing.

F Proof of Theorem 9

The problem stated in the theorem is reduced to the respective version of ε-Borsuk-Ulam parameterized by k in the same way as it was done in the proof of Theorem 6. Recall, k is the exact number of turns the SC-path should have. Suppose now we are given an instance as described in the statement of Theorem 9. Using exactly the same arguments as in the proof of Theorem 6, we can construct the Borsuk-Ulam function $f : S^{k+1} \mapsto \mathbb{R}^n$ in polynomial time. What remains to be shown in order to prove inclusion in NP is that if there is a solution to the ε-SC-Pizza-Sharing with k turns in the SC-path, then there exists a rational solution with bounded bit length.

As shown in the proof of Theorem 8, the function f we use (proof of Theorem 5) is continuous, piece-wise polynomial, and therefore λ-Lipschitz continuous, where $\lambda = \max_{j=1}^{n+1} \left\{ \sup_x \left|\left| \frac{\partial f(x)}{\partial x_j} \right|\right|_\infty \right\}$ (again, note that λ is constant, and points where $f_i(x)$, $i \in [n]$ is non-differentiable do not matter for Lipschitzness). Suppose that a point $x \in S^{k+1}$ satisfies $||f(x) - f(-x)||_\infty \leq \varepsilon$ for a given $\varepsilon > 0$, and therefore it is a solution to the decision version of ε-Borsuk-Ulam when parameterized by the number of turns k. Then, the following inequalities hold for any other point $y \in S^{k+1}$:

$$\begin{aligned} ||f(x) - f(-x)||_\infty &\leq \varepsilon, \\ ||f(x) - f(y)||_\infty &\leq \lambda \cdot ||x - y||_\infty, \\ ||f(-x) - f(-y)||_\infty &\leq \lambda \cdot ||x - y||_\infty. \end{aligned}$$

By the triangle inequality we get

$$||f(y) - f(-y)||_\infty \leq 2 \cdot \lambda \cdot ||x - y||_\infty + \varepsilon, \quad \text{for any} \quad y \in S^{k+1}. \tag{1}$$

Consider now a number M that is upper-bounded by an inverse-polynomial of the input size.

Then, for any y such that $||x-y||_\infty \leq \frac{M}{2\lambda}$ we have from (1) that

$$||f(y)-f(-y)||_\infty \leq M+\varepsilon,$$

meaning that y is also a solution to the ε'-Borsuk-Ulam parameterized by k, for $\varepsilon' = M+\varepsilon$. Therefore, y is a solution for the same problem when additively relaxed by at most an inverse polynomial (to the input) quantity.

By the above, we conclude that there are always polynomial-size solutions to the problem (and thus to the initial ε-SC-Pizza-Sharing problem) given that there are exact solutions to it. Therefore, a non-deterministic Turing machine can guess such a solution and verify that is indeed a solution in polynomial (to the input) time.

G Proof of Lemma 10

Proof. Consider a solution (R^-, R^+) for ε-SC-Pizza-Sharing with k cuts. For every $j \in [m]$ denote by R_j^- and R_j^+ the part of of square s_j that belong to regions R^- and R^+ respectively. For every square the path traverses, we will create either one or two cuts in I_{CH} depending on the way the path cuts this square. More formally, assume that square s_j is traversed by the SC-path. We will consider the following cases for the points (j,j) and $(j+1,j+1)$.

- Case 1: $(j,j) \in R^+$ and $(j+1,j+1) \in R^-$. Then, we will add one cut at $y = x_j + |R_j^+| \cdot (x_{j+1}-x_j)$ and the label below y will be "$+$" and above y "$-$".
- Case 2: $(j,j) \in R^-$ and $(j+1,j+1) \in R^+$. Then, we will add one cut at $x_j + |R_j^-| \cdot (x_{j+1}-x_j)$ and the label below y will be "$-$" and above y "$+$".
- Case 3: $(j,j) \in R^+$ and $(j+1,j+1) \in R^+$. Then, we will add one cut at x_j and one at $y = x_j + |R_j^+| \cdot (x_{j+1}-x_j)$ and the label for $[x_j, y]$ will be "$-$" and for $[y, x_{j+1}]$ will be "$+$".
- Case 4: $(j,j) \in R^-$ and $(j+1,j+1) \in R^-$. Then, we will add one cut at x_j and one at $y = x_j + |R_j^-| \cdot (x_{j+1}-x_j)$ and the label for $[x_j, y]$ will be "$+$" and for $[y, x_{j+1}]$ will be "$-$".

See Figure 6 for a visualisation of some of the cases.

Observe that the rules above create a valid labelling for $\mathcal{I}$; the labels alternate between the intervals induced by the cuts. In addition, observe that for any of the first two cases, with the exception of the first square the SC-path traverses, at least one turn required in order the SC-path to change direction and cross the square. Hence, we add at most one cut for any of these squares. In order one of the last two cases to occur, observe that the path needs to make at least two turns; one turn in order to enter the square and one turn in order to place point $(j+1,j+1)$ at the same region with the point (j,j). So, we add two cuts for any of these squares. Hence, the number of cuts we have created in I_{CH} is at most the same as the number of turns of the path plus one for the cut we have created for the first square. Hence we have at most $k+1$ cuts, since the SC-path has at most k turns.

In order to complete the proof of the lemma, we need to prove that this set of cuts is indeed a solution for I_{CH}. To see this, observe that for every $i \in [n]$ we have that

$$\mu_i(R^+) = \sum_j c_{ij} \cdot |R_j^+| = \sum_j c_{ij} \cdot |\mathcal{I}_j^+| = v_i(\mathcal{I}^+).$$

Using identical arguments, we get that $\mu_i(R^-) = v_i(\mathcal{I}^-)$. The lemma follows, since we have assumed that that (R^-, R^+) is a solution for ε-SC-Pizza-Sharing and thus $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon$ for every $i \in [n]$. □

H Checkerboard reduction

H.1 Construction

Our starting point is an instance I_{CH} of ε-Consensus-Halving with two-block uniform valuation and the definition of points of interest $0 \leq x_1 \leq x_2 \leq \ldots \leq x_m \leq 1$ as they are defined in the "overlapping reduction" paragraph. Recall, that in a two-block uniform valuation, every agent is associated with a number c_i that denotes the height of their blocks. Let $c_{\max} = \max_i c_i$. We will create an instance ε'-SC-Pizza-Sharing, I_{SCU}, where $\varepsilon' = \frac{\varepsilon \cdot n}{c_{\max}}$. In what follows, A_j will denote the set of agents that have positive value over the the interval of interest $[x_j, x_{j+1}]$. Furthermore, for every $i \in n$, let Z_i denote the set of intervals that agent i has positive value for.

Mass Distributions. For every agent $i \in [n]$ we will create a mass distribution μ_i which will consist of squares of size $\sqrt{c_i/n^2} \times \sqrt{c_i/n^2}$. In addition, μ_i will be uniformly distributed over each square.

Tiles. We will create $m-1$ types of tiles. A tile consists of a $n \times n$ square-grid where each square of the grid has size $\sqrt{c_{\max}/n^2} \times \sqrt{c_{\max}/n^2}$. We use A_j in order to define tile t_j; every row and every column of the $n \times n$ grid will contain exactly one square associated to mass μ_i. Hence, for every $i \in A_j$ tile t_j will contain n squares related to mass distribution μ_i. If $i \in A_j$, then the $(k, (i+k) \mod n)$th square will be associated with mass distribution μ_i, for every $k \in [n]$. The square of size $\sqrt{c_i/n^2} \times \sqrt{c_i/n^2}$ will be located at the centre of $(k, (i+k) \mod n)$th square where $k \in [n]$; formally, the bottom-left corner of this square will be located at $\left(k + \frac{\sqrt{c_{\max}/n^2} - \sqrt{c_i/n^2}}{2}, ((i+k) \mod n) + \frac{\sqrt{c_{\max}/n^2} - \sqrt{c_i/n^2}}{2}\right)$. See Figure 7(b) for a visualisation.

Blocks. We will create $m-1$ blocks $B_1, B_2, \ldots, B_{m-1}$. Block B_j will have edge-size $y_j := \sqrt{(x_{j+1} - x_j) \cdot c_{\max}}$ and will be tiled with $\frac{x_{j+1}-x_j}{\delta}$ copies of tile t_j. So, mass μ_i appears in B_j, only if $i \in A_j$. Observe that for every $i \in [n]$ and every $j \in [m-1]$ we have that:

- every row and every column of B_j contains exactly $\sqrt{n \cdot (x_{j+1} - x_j)}$ squares related to mass μ_i;
- B_j contains $n^2 \cdot (x_{j+1} - x_j)$ squares of size $\sqrt{c_i/n^2} \times \sqrt{c_i/n^2}$ each, associated with mass μ_i where $i \in A_j$;
- $\mu_i(B_j) = (x_{j+1} - x_j) \cdot c_i$ and $\mu_i(R) = \sum_{j \in Z_i} \mu_i(B_j) = 1$.

Next we explain the position of B_j on the plane; intuitively the bottom-left and the top-right corner of every block lie on the diagonal line $y = x$. Let $z_1 = 0$ and let $z_j := z_{j-1} + y_j$. The bottom-left corner of block B_j is at (z_j, z_j), i.e., block b_j spans over the points (z_j, z_j), (z_j, z_{j+1}), (z_{j+1}, z_j), and (z_{j+1}, z_{j+1}). Observe that this construction guarantees that any SC-path with k turns can cross at most $k+1$ blocks.

Observation 1. *For every $i \in [n]$, mass μ_i is associated with $\frac{n^2}{c_i}$ squares.*

Proof. Recall that μ_i is created from the valuation function of agent i: the valuation is two-block uniform where each block has height c_i and the total value of the agent for $[0,1]$ is 1. Thus, we get that $\sum_j (x_{j+1} - x_j) = \frac{1}{c_i}$. Then, the proof follows by observing that in every block contains $n^2 \cdot (x_{j+1} - x_j)$ squares of mass μ_i and then summing over all blocks. □

H.2 Proof of Theorem 14

In this section we use the construction of Section H.1 to prove Theorem 14.

Consider a solution for the instance I_{SCU} of ε'-SC-Pizza-Sharing. For every $j \in [m-1]$ denote by R_j^- and R_j^+ the part of block b_j that belongs to regions R^- and R^+ respectively. In addition, we will use $|R_j^+|$ and $|R_j^-|$ to denote the portion of block b_j that respectively belongs to R^+ and R^-. So, for every $j \in [m-1]$ we have $|R_j^+| \in [0,1]$ and $|R_j^-| \in [0,1]$ and $|R_j^+| + |R_j^-| = 1$. Since R^+ and R^- form a solution, we have $|\mu_i(R^+) - \mu_i(R^-)| \leq \varepsilon'$ for every $i \in [n]$. Recall, due to the way we have constructed I_{SCU}, any SC-path with k turns can cross at most $k+1$ blocks.

For every block B_j the SC-path crosses, we will create either one or two cuts at $[x_j, x_{j+1}]$ depending on the way the path crosses this block, using the same approach we used in the "overlapping reduction" paragraph. More formally, assume that block b_j is traversed by the SC-path. We will distinguish two cases depending on whether B_j is the fist block the SC-path crosses or not.

If B_j is the first block the path crosses and $(z_j, z_j) \in R^-$, then we will add one cut at $y = x_j + |R_j^+| \cdot (x_{j+1} - x_j)$ and the label below y will be "$+$" and above y it will be "$-$". Else if $(z_j, z_j) \in R^+$, then we will add one cut at $y = x_j + |R_j^+| \cdot (x_{j+1} - x_j)$ and the label below y will be "$-$" and above y it will be "$+$".

For every other block B_j the SC-path crosses, we will consider the following cases for the points (z_j, z_j) and (z_{j+1}, z_{j+1}).

1. $(z_j, z_j) \in R^+$ and $(z_{j+1}, z_{j+1}) \in R^-$. Then, we will add one cut at $y = x_j + |R_j^+| \cdot (x_{j+1} - x_j)$ and the label below y will be "$+$" and above y "$-$".
2. $(z_j, z_j) \in R^-$ and $(z_{j+1}, z_{j+1}) \in R^+$. Then, we will add one cut at $x_j + |R_j^-| \cdot (x_{j+1} - x_j)$ and the label below y will be "$-$" and above y "$+$".
3. $(z_j, z_j) \in R^+$ and $(z_{j+1}, z_{j+1}) \in R^+$. Then, we will add one cut at x_j and one at $y = x_j + |R_j^+| \cdot (x_{j+1} - x_j)$ and the label for $[x_j, y]$ will be "$-$" and for $[y, x_{j+1}]$ will be "$+$".
4. $(z_j, z_j) \in R^-$ and $(z_{j+1}, z_{j+1}) \in R^-$. Then, we will add one cut at x_{j-1} and one at $y = x_j + |R_j^-| \cdot (x_{j+1} - x_j)$ and the label for $[x_j, y]$ will be "$+$" and for $[y, x_{j+1}]$ will be"$-$".

We are using the same idea as before, hence see Figure 6 for a visualisation of some of the cases.

Observe that the rules above create a valid labelling for $\mathcal{I}$; the labels alternate between the intervals induced by the cuts. In addition, observe that for Cases 1 and 2, with the exception of the first block the SC-path traverses, at least one turn required in order the SC-path to change direction and cross the block. Hence, we add at most one cut for any of these blocks. In order one of Cases 3 and 4 to occur, observe that the path needs to make at least two turns; one turn in order to enter the block and one turn in order to place point (z_{j+1}, z_{j+1}) at the same region with the point (z_j, z_j). So, we add two cuts for any of these blocks. Hence, the number of cuts we have created is at most the same as the number of turns of the path plus one for the cut we have created for the first block. Hence if the path has k turns, we have at most $k + 1$ cuts.

In order to complete the proof, we need to prove that this set of cuts is indeed an ε-solution for I_{CH}. To do so, we fist need to define some useful quantities. Consider the squares of size $\sqrt{c_i/n^2} \times \sqrt{c_i/n^2}$ that define mass distribution μ_i. Let S_i denote this set of squares and let S_{ij} denote the subset of S_i that belong to block B_j, i.e. $S_i = \cup_{j \in [m-1]} S_{ij}$. Denote S_j^+ the squares that belong to R^+; S_i^- the squares that belong to R^-; and S_i^c the squares that are crossed from the SC-path so they contain parts of both R^+ and R^-. In addition, denote S_{ij}^+, S_{ij}^-, and S_{ij}^c the subsets of S_i^+, S_i^-, and S_i^c constrained on block b_j.

The uniform spread of the mass distributions over block B_j guarantees that if $|R_j^+| = p_j \in [0, 1]$, then for every $i \in A_j$ it holds that

$$\frac{|S_{ij}^+| - |S_{ij}^c|}{|S_{ij}|} \leq p_j \leq \frac{|S_{ij}^+| + |S_{ij}^c|}{|S_{ij}|}. \tag{2}$$

The following observations will be useful in our proof.

Observation 2. *Assume that there are k_j turns in block B_j. Then for every $i \in [n]$, we have that $|S_{ij}^c| \leq (k_j + 1) \cdot \sqrt{n \cdot (x_{j+1} - x_j)}$.*

Proof. Recall that block B_j if a grid of rows and columns of squares of size $\sqrt{c_{\max}/n^2} \times \sqrt{c_{\max}/n^2}$. Observe that every row of B_j contains $n \cdot (x_{j+1} - x_j)$ squares associate to mass μ_i. An SC-path with k turns can cross at most $(k + 1)$ rows or columns of B_j, hence the correctness follows. □

In every solution for ε'-SC-Pizza-Sharing for I_{SCU} we the the following.

Lemma 20. *In every solution for ε'-SC-Pizza-Sharing it holds that* $\left|\sum_{j\in Z_i} p_j \cdot |S_{ij}| - |S_i|/2\right| \leq \varepsilon' \cdot |S_i| - 5n^{3/2}$.

Proof. We will prove the lemma by contradiction. So, for the sake of contradiction, assume that there exists a solution for ε'-SC-Pizza-Sharing such that for some $i \in [n]$ we have that $\left|\sum_{j\in Z_i} p_j \cdot |S_{ij}| - |S_i|/2\right| > \varepsilon' \cdot |S_i| - 5n^{3/2}$. Without loss of generality assume that

$$\sum_{j\in Z_i} p_j \cdot |S_{ij}| > |S_i|/2 + \varepsilon' \cdot |S_i| - 5n^{3/2}. \tag{3}$$

Then we have the following:

$$\begin{aligned}
\mu_i(R^+) &= \sum_{j\in Z_i} \mu_i(R^+ \cap b_j) \\
&\geq \frac{c_i}{n^2} \cdot \sum_{j\in Z_i} |S_{ij}^+| \\
&\geq \frac{c_i}{n^2} \cdot \sum_{j\in Z_i} \left(p_j \cdot |S_{ij}| - |S_{ij}^c|\right) \quad \left(\text{since from (2) we have that } |S_{ij}^+| \geq p_j \cdot |S_{ij}| - |S_{ij}^c|\right) \\
&\geq \frac{c_i}{n^2} \cdot \sum_{j\in Z_i} \left(p_j \cdot |S_{ij}| - (k_j+1) \cdot \sqrt{n \cdot (x_{j+1} - x_j)}\right) \quad \text{(from Observation 2)} \\
&\geq \frac{c_i}{n^2} \cdot \sum_{j\in Z_i} \left(p_j \cdot |S_{ij}| - (k_j+1) \cdot \frac{1}{\sqrt{n}}\right) \\
&\geq \frac{c_i}{n^2} \cdot \left(\sum_{j\in Z_i} p_j \cdot |S_{ij}| - 5n^{3/2}\right) \quad \left(\text{since } \sum_j k_j \leq n \text{ and } |Z_i| \leq 4n\right) \\
&> \frac{c_i}{n^2} \cdot \left(|S_i|/2 + \varepsilon' \cdot |S_i|\right) \quad \text{(from Eq. (3))} \\
&= \frac{1}{2} + \varepsilon' \quad \left(\text{since } |S_i| = \frac{n^2}{c_i}\right) \quad \text{(from Observation 1).}
\end{aligned}$$

This contradicts the assumption that R^+ is part of a solution, since we know that $\mu_i(R) = 1$ and from the inequality above we would have $\mu_i(R^+) - \mu_i(R^-) > \varepsilon'$. To complete the proof we have to consider the case where $\sum_{j\in Z_i} p_j \cdot |S_{ij}| < |S_i|/2 - \varepsilon' \cdot |S_i| + 5n^{3/2}$ instead of (3). Then, using identical arguments as above, but using this time the left part of (2), we get that $\mu_i(R^+) < \frac{1}{2} - \varepsilon'$ which will contradict again the assumption that R^+ is part of a solution. □

Now we are ready to prove the correctness of our theorem.

Lemma 21. *For the constructed solution for* Consensus-Halving *instance I_{CH}, we have that that $|v_i(\mathcal{I}^+) - v_i(\mathcal{I}^-)| \leq \varepsilon$, for every $i \in [n]$.*

Proof. Observe that our translation of the SC-path to cuts implies that

$$v_i(\mathcal{I}^+) = \sum_j p_j \cdot c_i \cdot (x_{j+1} - x_j) \quad \text{and} \quad v_i(\mathcal{I}^-) = \sum_j (1 - p_j) \cdot c_i \cdot (x_{j+1} - x_j).$$

So, we get that

$$
\begin{aligned}
|v_i(\mathcal{I}^+) - v_i(\mathcal{I}^-)| &= \left| 2 \cdot \sum_{j \in Z_i} p_j \cdot c_i \cdot (x_{j+1} - z_j) - \sum_{j \in Z_i} c_i \cdot (x_{j+1} - z_j) \right| \\
&= \frac{c_i}{n^2} \cdot \left| 2 \cdot \sum_{j \in Z_i} p_j \cdot |S_{ij}| - \sum_{j \in Z_i} |S_{ij}| \right| \\
&= \frac{c_i}{2n^2} \cdot \left| \sum_{j \in Z_i} p_j \cdot |S_{ij}| - \frac{|S_i|}{2} \right| \\
&\leq \frac{c_i}{2n^2} \cdot \left(\varepsilon' - 5n^{3/2} \right) \qquad \text{(from Lemma 20)} \\
&< \frac{c_i}{2n^2} \cdot \varepsilon' \\
&\leq \varepsilon \qquad \left(\text{since } \varepsilon' = \frac{\varepsilon \cdot n}{c_{\max}} \right).
\end{aligned}
$$

□

I Proof of Lemma 15

Proof. Point 3 from the I_{CH}^{DFMS} instance guarantees the following for the produced instance of SC-Pizza-Sharing. For every mass distribution μ_i there exist either three consecutive unit-squares s_j, s_{j+1}, s_{j+2} (or four consecutive unit-squares s_j, s_{j+1}, s_{j+2}, s_{j+3}) such that $\mu_i(s_j \cup s_{j+1} \cup s_{j+2}) > \frac{1}{2} \cdot \mu_i(\mathbb{R}^2)$ (or $\mu_i(s_j \cup s_{j+1} \cup s_{j+2} \cup s_{j+3}) > \frac{1}{2} \cdot \mu_i(\mathbb{R}^2)$) and in addition for every $i \neq i'$ we have $\mu_{i'}(s_j \cup s_{j+1} \cup s_{j+2}) = 0$ (or $\mu_{i'}(s_j \cup s_{j+1} \cup s_{j+2} \cup s_{j+3}) = 0$). Hence, in every solution of SC-Pizza-Sharing the SC-path has to cross every such triplet (or quadruple) of unit-squares. But since these triplets (or quadruples) are not overlapping for any pair of mass distributions and we need to have at most $n - 1$ turns, we cannot have any turn of the SC-path inside these triplets of unit-squares. If the SC-path had such a turn, then inevitably it could not cross every triplet of unit-squares hence it could not be a solution for SC-Pizza-Sharing. Since there are no turns inside the triplets of unit-squares and since the unit-squares lie on the diagonal $x = y$, then there is no turn inside any unit-square. □

J Proof of Theorem 17

Before we prove the theorem, let us give a brief sketch of the ETR-hardness proof of the aforementioned Consensus-Halving decision version of [7]. We are given an instance of the following problem which was shown to be ETR-complete (Lemma 15, [7]).

Definition 22 (Feasible$_{[0,1]}$)**.** *Let $p(x_1, \ldots, x_m)$ be a polynomial. We ask whether there exists a point $(x_1, \ldots, x_m) \in [0,1]^m$ that satisfies $p(x_1, \ldots, x_m) = 0$.*

Given the polynomial p, we first normalize it so that the sum of the absolute values of its terms is in $[0, 1]$ (thus not inserting more roots), resulting to a polynomial q. Then, we separate the terms that have positive coefficients from those that have negative coefficients, thus creating two positive polynomials q_1, q_2 such that $q = q_1 - q_2$. Therefore, $p(\vec{x}) = 0$ for some $\vec{x} \in [0,1]^m$ if and only if $q_1(\vec{x}) = q_2(\vec{x})$. We then represent q_1, q_2 in a circuit form with gates that implement the operations $\{c, +, \times c, \times\}$ (where $c \in [0,1] \cap \mathbb{Q}$ is a constant input, and $\times c$ is multiplication by constant). By the scaling we know that $q_1, q_2 \in [0,1]$, and in addition, the computation of the circuit using the aforementioned operations can be simulated by a Consensus-Halving instance with $n - 1$ agents, where $n - 1 \in \mathit{poly}(\#\mathit{gates})$; the argument of p becomes a set of "input" cuts and according to the circuit implementation by Consensus-Halving, two output cuts encode q_1 and q_2. Finally, checking whether $q_1 = q_2$ is true is done by an additional (n)-th agent that can only be satisfied (have her total valuation split in half) if

and only if $q_1(\vec{x}) = q_2(\vec{x})$ for some $\vec{x} \in [0,1]^m$. In other words, the Consensus-Halving instance has a solution if and only if there are "input" cuts $\vec{x} = (x_1, \ldots, x_m) \in [0,1]^m$ that force the rest of the cuts (according to the circuit implementation) that encode the values $v_{m+1}, \ldots, v_{n-1}$ at the output of each of the circuit's gates such that they also cut the (n)-th agent's valuation in half without the need for an additional cut.

We are now ready to prove the theorem.

Proof. We will use exactly the same technique up to the point where we have a Consensus-Halving instance that checks whether $q_1 = q_2$. Then, we use the gadgets described in the FIXP-hardness reduction of Section 4.2 that reduce the valuation functions of n agents in Consensus-Halving into mass distributions of a SC-Pizza-Sharing instance with n colours. According to Lemma 15 in any solution of the resulting SC-Pizza-Sharing instance, the SC-path does not have turns inside any unit-square. This means that each horizontal/vertical segment of the SC-path that cuts a unit square in SC-Pizza-Sharing has a 1-1 correspondence to a cut of a Consensus-Halving solution, thus a Consensus-Halving solution that uses $n-1$ cuts would correspond to a SC-path with $n-1$ line segments, i.e. $n-2$ turns. Therefore, if and only if there is a SC-path that solves SC-Pizza-Sharing with n colours and $n-2$ turns, there is a $(n-1)$-cut that solves Consensus-Halving with n agents, and equivalently there is a $\vec{x} \in [0,1]^m$ such that $p(\vec{x}) = 0$, making the $\text{Feasible}_{[0,1]}$ instant "yes". □